

UNIVERSITY OF GENOVA

POLYTECHNIC SCHOOL

DIME

Department of Mechanical, Energy, Management
and Transportation Engineering



MASTER OF SCIENCE THESIS

IN

MECHANICAL ENGINEERING

**Numerical Simulations on Atmospheric Chamber
(ChAMBRé) to study Bioaerosol**

SUPERVISOR:

Chiar.^{mo} Prof. Ing. Jan Oscar Pralits (Unige)

CO SUPERVISOR:

Dott. Marco Atzori (KTH)

CANDIDATE:

Giovanni Curredda

March 2019

Numerical Simulations of the Atmospheric Chamber (ChAMBRe) to Study Bioaerosol

Giovanni Curredda

supervised by

Prof. Ing. Jan Oscar Pralits (Unige)

Dott. Marco Atzori (KTH)

in collaboration with INFN

M. Brunoldi, S.G. Danelli, D. Massabó, F. Parodi, P. Prati, V. Vernocchi

Abstract

Atmospheric chambers are experimental facilities made to reproduce real environments and to study the interactions between their constituents. An atmospheric chamber, named ChAMBRe is installed at the National Institute of Nuclear Physics (INFN) in Genoa. ChAMBRe is specifically designed for the research on atmospheric bio-aerosol. The presented work concerning this facility starts the beginning of a collaboration with the INFN. The aim of the thesis is to study the fluid dynamics of ChAMBRe using OPENFOAM[®], an open-source CFD software. An initial part describes the atmospheric chamber and its 3D CAD model. Afterwards several numerical simulations are outlined: from the airflow study to the multiphase CFD with particles. The final purpose consists in improving of experimental facility, a task that can't be realized without a deep understanding of its physics.

Simulazioni Numeriche della Camera Atmosferica (ChAMBRé) per Studiare Bioaerosol

Giovanni Curredda

Con la supervisione di

Prof. Ing. Jan Oscar Pralits (Unige)

Dott. Marco Atzori (KTH)

In collaborazione con INFN

M. Brunoldi, S.G. Danelli, D. Massabó, F. Parodi, P. Prati, V. Vernocchi

Sommario

Le camere atmosferiche sono installazioni sperimentali realizzate per riprodurre ambienti reali e studiare le interazioni tra i loro costituenti. Una camera atmosferica, chiamata ChAMBRé, è situata presso l'Istituto Nazionale di Fisica Nucleare (INFN) a Genova. ChAMBRé è progettata per svolgere ricerche sui bio-aerosol atmosferici. Il lavoro qui presentato riguarda questa struttura sperimentale segna l'inizio di una collaborazione con l'INFN. L'obiettivo della tesi è lo studio della fluidodinamica di ChAMBRé tramite OPENFOAM[®], un software CFD gratuito. Successivamente alla realizzazione di un modello CAD 3D della camera, diverse simulazioni numeriche sono state implementate: a partire dallo studio del flusso d'aria fino alla fluidodinamica multifase con particelle. Il fine ultimo consiste nel migliorare questa installazione sperimentale, traguardo irrealizzabile senza una profonda conoscenza della sua fisica.

Acknowledgements

Ringrazio il Prof. Ing. Jan Pralits per l'opportunità e il sostegno offerti durante questo lavoro di tesi.

Ringrazio il Dott. Marco Atzori, per il suo contributo preciso e fondamentale.

Ringrazio tutti i membri dell'INFN per questa serena e fruttuosa collaborazione.

Ringrazio i miei amici, compagni di avventure.

Ringrazio i miei genitori, costante conforto.

Ringrazio fratello Francesco, instancabile guida.

Ringrazio Noemi che è al mio fianco.

Contents

1	Introduction	1
I	Case Study and Numerical Tools	2
2	ChAMBRé description	3
2.1	CAD geometry	4
3	Open Source Tools	8
3.1	Onshape	8
3.2	OpenFOAM	8
3.3	OpenFOAM case structure	11
3.4	The programming language of OpenFoam	12
3.4.1	Why <i>C++</i>	12
II	Fluid Dynamics Characterization of ChAMBRé	14
4	Governing Equations	15
4.1	Conservation laws	15
4.1.1	Continuity equation	15
4.1.2	Conservation of Linear Momentum	16
4.1.3	Conservation of Energy	17
4.2	Turbulence modelling	18
4.3	Shear Stress Transport $k - \omega$ Model	21
4.4	Near the wall treatment	23
5	RANS Simulations with MRF Method	25
5.1	<i>Coarse Mesh</i> generation	25
5.2	Multiple Reference Frame Method	29

5.3	Results	30
5.3.1	<i>noPlate</i> simulations	30
5.3.2	<i>Plate</i> simulations	39
5.4	<i>Plate</i> and <i>noPlate</i> cases comparison	47
6	URANS Simulation with Sliding Mesh Method	50
6.1	<i>Fine Mesh</i> generation	51
6.2	Sliding Mesh Method	54
6.3	RANS with <i>Fine Mesh</i>	54
6.4	Results	57
III	Simulations with Particles	59
7	Particle Implementation	60
7.1	Governing Equations	60
7.1.1	Particle equations	61
7.2	Dispersion Models	63
7.3	Case study, Choice and Setup	63
8	Particle Simulations and Results	65
8.1	Particles <i>noPlate</i> results	66
8.2	Particles <i>Plate</i> results	68
	Conclusions	73

Chapter 1

Introduction

Atmospheric chambers are a rising experimental facility made for research purposes. Their experimental setup varies with their applications but the basic idea is to reproduce environmental condition in a controlled volume. The fields of research include biology, industrial products and materials. In Europe, EUROCHAMP-2020 consortium is a project that joins various chambers that research in air quality and climate models [1]. An atmospheric chamber, named ChAMBRé, is part of the consortium and is located in Genoa at the Department of Physics. The National Institute of Nuclear Physics INFN designed and installed this facility in January 2017. ChAMBRé is specifically designed for the research on atmospheric bio-aerosols which are suspensions of fine solid particles or liquid droplets in air. The diameter of the particles injected in the chamber varies from few hundreds of nanometers to a few microns. Characteristic parts of the facility are the equipment and the procedures to grow, inject and extract bacterial strains in the chamber volume [2]. Preserving the bacteria's viability is a fundamental task for ChAMBRé and the INFN exposed the need of an improvement for this feature. The lifetime of bacteria is directly related with the wall losses [3]. Phenomena of particles deposition can be simulated with computational fluid dynamics. This led to the beginning of a collaboration between two departments: DICCA and DIFI. The common aim is to modelize, study and propose solutions and improvements for ChAMBRé.

Part I

Case Study and Numerical Tools

Chapter 2

ChAMBRé description

The dissertation starts with a description of ChAMBRé which will be focused on the components that are fundamental for a CFD study of the facility. The structure essentially is a stainless steel vessel of cylindrical shape with domed bases. It has maximum height and diameter respectively of 2.9 m and 1 m with a total volume of about 2.23 cubic meters. The main body is divided into three parts: two domed cylinders connected by a central ring. The whole structure is maintained in vertical position by an ad-hoc metallic structure with plastic supports that damps vibration phenomena. The vessel has several flanges that are made for various purposes. Four flanges can be set up for many measurement instruments. Concentration, humidity and other measurements are made by spilling a mass flow from the chamber. Once the flow passes through the instruments, all the spilled quantities are recollected and reinjected by a single flange at the low part of the vessel. One flange is dedicated at the cleaning of the ChAMBRé with a vacuum pump, this part is not considered in the model because it will be closed and inactive during the experimental tests.

The central ring allocates the most important flange for our study: the injection flange. Through this opening, a volume of about 3 milliliters of cells suspension is sprayed into the chamber using a Blaumstein single jet Atomizer. The atomizer is specifically designed to provide bio-aerosols with high viability of microorganisms for aerobiology research. The viability is due to the process that utilizes minimal energy to properly aerosolize the liquid. The atomizing head is composed of two main parts: nozzle body and expansion plate. Pressurized air at 3.8 bar passes at sonic velocity through a hole made with a laser cut ruby crystal. The liquid with particles is carried into a cavity between the nozzle body and the expansion plate with a precision syringe pump. The aerosol is generated when the air jet breaks up the liquid into droplets. Then the mixture is forced to follow a U-turn in order

to separate the bigger particles. The result is a very fine mist with narrow size distribution that will approach inside the chamber passing first through a curved stainless-steel tube then through a flange orifice of 25 millimeters.

To favour the mixing of the gas and aerosol species in the reactor a fan is installed in the bottom part of the chamber. It is a standard venting system with six metallic blades of 25 centimeters length each connected to an electrical engine through a rotating shaft. A particular pass through has been designed and built at INFN (National Institute of Nuclear Physics) to ensure the vacuum seal. The fan rotating speed can be regulated from 0 to 10 Hz.

Above the fan there is a second, fundamental element of the facility: a thin rectangular plate. The plate inserted in a central position inside the chamber through a specific designed flange that allows this element to translate on trolley. The purpose of this system is to collect and extract bacteria or particles during the tests, in fact six petri capsules are installed on the plate. During an experiment of bacteria the plate flange will be hermetically close by special pneumatic valves and the plate will be placed inside the chamber in order to collect bacteria. At the end of the test the plate will be extracted from the chamber and the petri capsules, closed by a complex leverage system, will be further analyzed. The plate is not always needed during experiments hence the need of investigating the flow with and without the plate at the centre of ChAMBRé.

A 90 cm long UV lamp is permanently installed inside the chamber. The lamp is inserted through the flange in the top dome and it produces UV radiation which is used to sterilize, without producing ozone, the chamber volume, in particular after any experiment with bio-aerosol.

Fig.(2.1) presents a schematic view of ChAMBRé.

2.1 CAD geometry

The first basic step in a computational fluid dynamics case is to obtain the geometry of the domain where we will run our simulations. The following section describes all the assumptions that has been made when going from the reality to the aerodynamic model of ChAMBRé. All the following steps have been carried out using an online opensource CAD software: Onshape. The basic guideline in this process of geometrical modelling is to keep the model as close as possible to the reality without forgiving that computational fluid dynamics can be extremely time consuming. In order to achieve an acceptable computational effort simplifications of

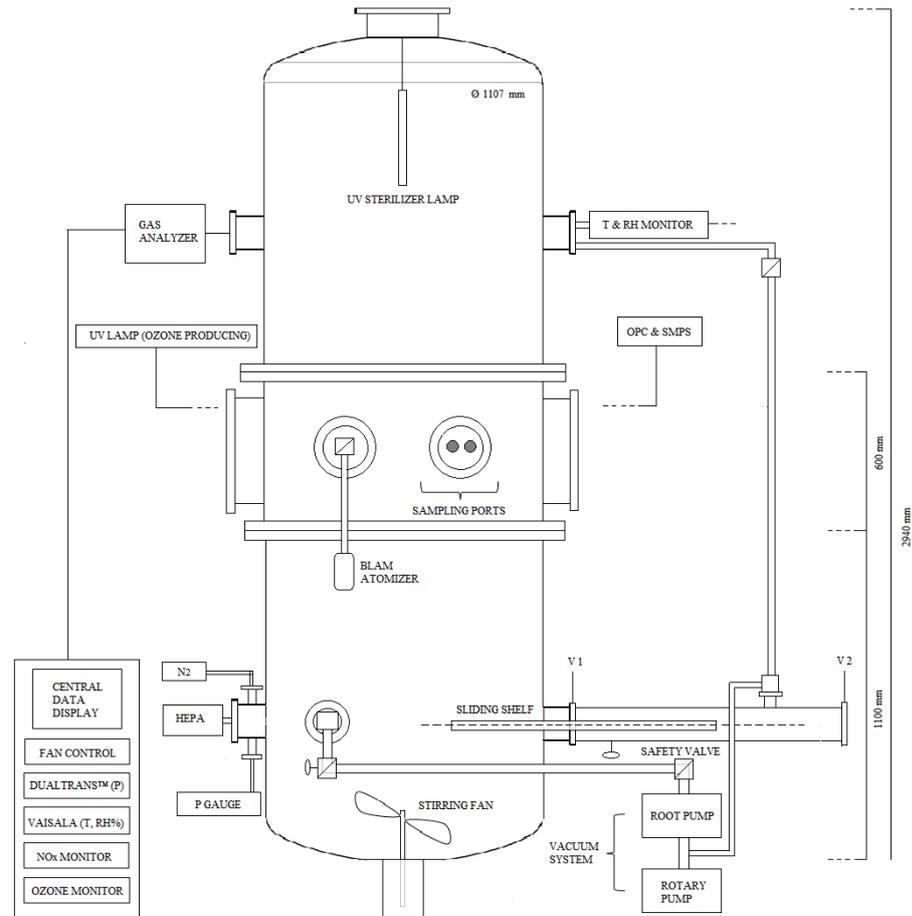
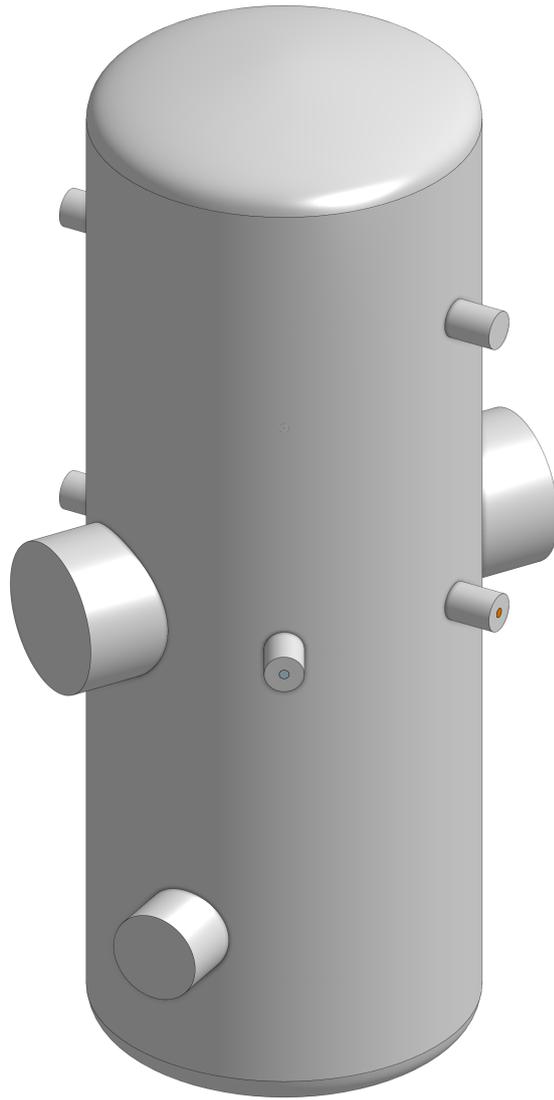


Figure 2.1: *ChAMBRé working scheme*

the real geometry are needed. If the physics of the problem (and the experience in CFD) suggests that an element isn't fundamental in the overall aerodynamic of the chamber, the element can be simplified. The Cad design started from an existing 3D geometry produced by the mechanical design service of the INFN. The chamber vessel was represented with high fidelity but we decided to redesign it in order to have deep knowledge of the model and above all to ensure full compatibility with our softwares. Inside the chamber the sophisticated shaft seal could be assumed as wall with no losses of volume. The shaft of the fan and the UV lamp can be eliminated in the model because they will not greatly affect the fluid motion and even if they will, the axisymmetry of the model permits to consider their effect negligible. The Fan is a fundamental part of the study because it is used to achieve mixing of the particles in the chamber. It is known that the shape of the six blades of the fan will generate a precise motion of the fluid. Differences from the real shape of the object can produce big mistakes in the motion generated by the reconstructed geometry

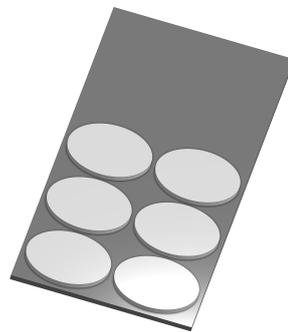
in the computational ambient. Unfortunately there was no 3D CAD model of this component that requires an high fidelity representation. Reconstructing the fan geometry was not trivial and implied two different contributes: measurements made in the INFN's measure laboratory of the mechanical design service and drawings gently shared by the producer Elektrovent. The measure laboratory worked on a copy of the installed fan with a mechanical probe DEA Mistral 070705 that can measure the position of a point of a solid object and give as output three spatial coordinates. The data has been obtained by measuring position of points on the pressure side and suction side of the profile at several fixed distances from the center (i.e. radius). The measured points on the chord were allocated in such a way that were the curvature increases, the required number of measures increases. As a result many points were taken at leading edge and trailing edge of each profile. Once the coordinates has been wrote on a data file, a OnShape programmable feature were used to plot points on the 3D space. Successively each point of a single section has been connected by spline lines. At this step seven section of the entire blade were reconstruced. Three other measurement were done: the points distribution over the leading edge, trailing edge and over the tip of the blade. From the mean of the distribution of such points three straight lines were obtained. This allowed to join the sections following a closed path defined by these three straight lines. Once the blade has been reconstructed, the complex hub's geometry was simplified to a cylinder and six blades have been extruded on this cylinder obtaining the definitive fan 3D CAD ready to be implemented in the computational environment. During every step of this procedure the 2D drawings from the factory were used as benchmark comparing the measures of the drawings with the developing 3D geometry. As can be noted in the previous description, the rectangular plate is really complex and its geometry includes a lot of small details. This kind of details will not have a strong impact on the fluid motion but they can highly increase the computational cost of simulations and for this reason they will be simplified. The resulting geometry is a flat plate that contains six petri capsules represented as cylinders evenly spaced along the upper rectangular surface. All components designed are showed in Fig.(2.2).



(a) Chamber



(b) Fan



(c) Plate

Figure 2.2: *Components of ChAMBRe CAD designed*

Chapter 3

Open Source Tools

3.1 Onshape

Onshape is a computer-aided design (CAD) software system that is used in a wide range of industrial fields including consumer electronics, mechanical machinery, medical devices and 3D printing. It makes extensive use of cloud computing, every change in the geometry is saved and can be recovered at any time and multiple users can work online at the same project. The 3D modelling part of Onshape is completely free and online. It has a user friendly interface that allows also unexperienced users to approach and learn the CAD modeling. In this thesis this tool has been used for design the geometry of the case. Onshape support also a programming language, Feature Script that can be used to define custom feature. Various users write and share their own programmed features. A custom feature has been used in the reconstruction of the fan geometry [4].

3.2 OpenFOAM

OPENFOAM[®] (*Open source Field Operation And Manipulation*) is an open source finite volume software for computational fluid dynamics (CFD), owned by the OPENFOAM[®] Foundation and distributed exclusively under the GNU General Public Licence (GPL)[5]. That means it is freely available and according to the GNU general public license principles the users can modify and share the source code that is freely distributed. Generally speaking OPENFOAM[®] is a C++ library, used to solve partial differential equations (PDEs), and ordinary differential equations (ODEs). Its primary usage is to create executables, known as applications that fall into two categories: solvers, that are each designed to solve a specific problem in

continuum mechanics, and utilities, that are designed to perform tasks that involve data manipulation[5]. The OPENFOAM[®] distribution has an extensive range of features to solve anything from complex fluid flows involving combustion and chemical reactions, multiphase flows and mass transfer, turbulence and heat transfer, particle methods (DEM,DSMC,MD) and lagrangian particles tracking to acoustics, solid mechanics and electromagnetics. It includes tools for meshing in and around complex geometries, and for data processing and visualisation, and more. Almost all computations can be executed in parallel as standard to take full advantage of today's multi-core processors and multi-processor computers. OPENFOAM[®] is supplied with pre- and post-processing environments. The interface to the pre- and post-processing are themselves OpenFOAM utilities, thereby ensuring consistent data handling across all environments. [5]. The overall structure of OPENFOAM[®] is shown in *Figure 3.1*.

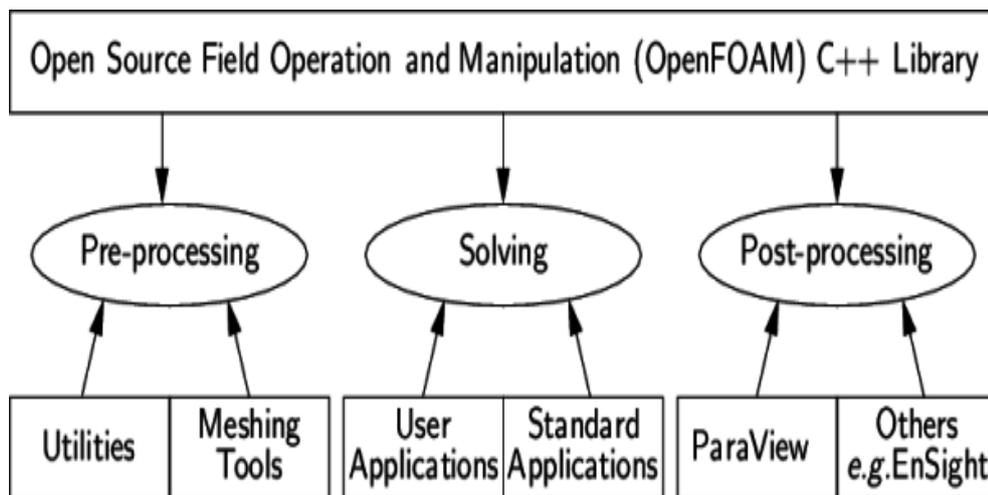


Figure 3.1: *Overview of OpenFOAM structure [5].*

Hence it is clear that OPENFOAM[®] capabilities mirror those of commercial *CFD* applications, but there are still some disadvantages compared to them, such as the lack of a native *GUI*, not much available documentation and, in wider terms, it is less user friendly; however, as the users have complete access to the source code, they have total freedom to modify existing solvers or use them as the starting point for new ones with some pre-requisite knowledge of the underlying method, physics and programming techniques involved. Summing up some of the features of OPENFOAM[®] are listed below taken from the official web-site [5]:

- FLUID DYNAMICS & PHYSICAL MODELLING

- Turbulence modelling (Reynolds-Averaged (RANS), Large-Eddy Simulation (LES), Detached-Eddy Simulation (DES,DDES,etc))
- Thermophysical modelling
- Transport/rheology
- Multiphase flows
- Rotating flows with multiple reference frames (MRF)
- Rotating flows with arbitrary mesh interface (AMI)
- Dynamic meshes
- Compressible/thermal flows
- Conjugate heat transfer
- Porous media
- Lagrangian particle tracking
- Reaction kinetics/chemistry
- GEOMETRY & MESHING
 - Mesh generation for complex geometries with *snappyHexMesh*
 - Mesh generation for simple geometries with *blockMesh*
 - Mesh conversion tools
 - Mesh manipulation tools
- NUMERICAL SOLUTION
 - Numerical method
 - Linear system solvers
 - Ordinary Differential Equation system solvers
- COMPUTING & PROGRAMMING
 - Equation syntax
 - Libraries of functionality
 - Parallel computing
- DATA ANALYSIS
 - *ParaView* post-processing

- Post-processing command line interface (CLI)
- Graphs and data monitoring

3.3 OpenFOAM case structure

The basic directory structure for a OPENFOAM[®] case, with the minimum set of files to run an application, is presented in *Figure 3.2*.

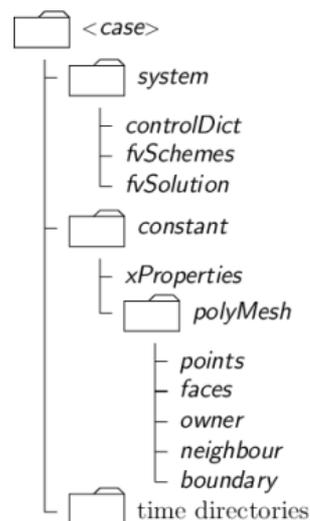


Figure 3.2: *Structure of an OPENFOAM[®] case [5].*

The roles of the main directories, contained in the case folder, are listed below:

- **system**, it contains the dictionaries to set up the entire solution procedure (from meshing to solving); at least it must contain three files:
 - **fvSchemes** to specify (run-time) the numerical schemes to discretize the equations;
 - **fvSolution** to set equation solvers, tolerances and other algorithm controls;
 - **controlDict** to control (run-time) the simulation run (start/end time, time-step, function objects etc.)
- **constant**, it contains a folder (**polyMesh**) with the full description of the case mesh and files that specify the physical properties involved (transport and turbulence properties, gravity, dynamic properties etc.)

- **time directories**, it contains files that represent the specific fields at initial condition (e.g. 0 folder) or computed by OPENFOAM[®] (e.g. 0.01, 0.02, ... folders¹) at consecutive times; it must be underlined that OPENFOAM[®] always require fields to be initialized, even in steady-state problems

A lot of pages should be written to exhaustively explain OPENFOAM[®], but that is beyond the scope of this thesis. For further details the *CFD direct* website [5] is suggested.

3.4 The programming language of OpenFoam

3.4.1 Why C++

C++ is an *Object-Oriented Programming* language that attempts to provide techniques for managing enormous complexity, achieving the aim of the reuse of software components. As an *Object-Oriented Programming* languages is based on three pillars of the object-oriented development [15]:

- encapsulation
- inheritance
- polymorphism.

Encapsulation

C++ supports the properties of encapsulation through the creation of user-defined types, called classes. Once created a well defined class acts as a fully encapsulated entity and it is used as a whole unit. The actual inner workings of the class should be hidden. Users of a well defined class do not need to know how the class works; they just need to know how to use it.

Inheritance

C++ supports inheritance; a new type (class), which is an extension of an existing type, can be declared. This new subclass is said to derive from the existing type (sometimes is called a derived type) and inherits all its qualities, but the user can add new ones as needed.

¹The name of the folder corresponds to the simulated time at which data are written.

Polymorphism

C++ supports the idea that different objects (belonging to the same class) do "the right thing" when the user chooses one of them. Being more exhaustive, in programming languages, polymorphism means that some code or operations or objects behave differently in different contexts.

A clarifier example inherent to a CFD code is relative to a velocity field. The expression encapsulates the idea of movement with direction and magnitude and relates to other physical properties. In mathematics, we can represent a velocity field by a single symbol, e.g. U , and express certain concepts using symbols, e.g. "the field of velocity magnitude" by $|U|$. The advantage of mathematics over verbal language is its greater efficiency, making it possible to express complex concepts with extreme clarity. The problems that we wish to solve in continuum mechanics are not presented in terms of intrinsic entities, or types, known to a computer, e.g. bits, bytes, integers. They are usually presented first in verbal language, then as partial differential equations in 3 dimensions of space and time. The equations contain the following concepts: scalars, vectors, tensors, and fields thereof; tensor algebra; tensor calculus; dimensional units. The solution to these equations involves discretisation procedures, matrices, solvers, and solution algorithms. Programming languages that are *Object-Oriented*, as stated in the introduction to this chapter, provide the mechanism to declare types -*classes*- and associated operations that are part of the verbal and mathematical languages used in science and engineering. The velocity field introduced earlier can be represented in programming code by the symbol U and "the field of velocity magnitude" can be $mag(U)$. The velocity is a vector field for which there should exist, in an *Object-Oriented*, a *vectorField* class. The velocity field U would then be an instance, or object, of the *vectorField* class; The clarity of having objects in programming that represent physical objects and abstract entities should not be underestimated. The class structure concentrates code development to contained regions of the code, the classes themselves, thereby making the code easier to manage. New classes can be derived or inherit properties from other classes, e.g. the *vectorField* can be derived from a *vector* class and a *Field* class. *C++* provides the mechanism of template classes such that the template class *Field*<*Type*> can represent a field of any <*Type*>, e.g. scalar, vector, tensor. The general features of the template class are passed on to any class created from the template. Templating and inheritance reduce duplication of code and create class hierarchies that impose an overall structure on the code [14].

Part II

Fluid Dynamics Characterization of ChAMBR_e

Chapter 4

Governing Equations

4.1 Conservation laws

The principle of conservation states that for an isolated system certain physical measurable quantities are conserved over a local region. This conservation principle or conservation law is an axiom that cannot be proven mathematically but can be expressed by a mathematical relation. Laws of this type govern several physical quantities such as mass, momentum, and energy (the Navier-Stokes equations)[13]. The conservation laws involving fluid flow and related transfer phenomena can be mathematically formulated following either a Lagrangian (material volume, MV) or an Eulerian (control volume) approach. Assuming a continuous phase the most common method of describing fluid flow is the fixed reference system Eulerian approach that is synthetically presented below. A short description of the Lagrangian method will be introduced in the next section.

4.1.1 Continuity equation

The principle of conservation of mass indicates that in the absence of mass sources and sinks, a region will conserve its mass on a local level [13]. Being ρ the density, through the application of the Reynolds transport theorem, the general expression for conservation of mass as applied to a control volume will be:

$$\int_V \frac{\partial \rho}{\partial t} dV + \int_S \rho \mathbf{V} \cdot \mathbf{n} dS = 0, \quad (4.1)$$

where S is the surface of the control volume V . Thanks to the divergence theorem – and noticing that the conservation of mass should be respected for every control volume – this equation can be written in a differential form, called the continuity

equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0 \quad (4.2)$$

4.1.2 Conservation of Linear Momentum

Through application of the Reynolds transport theorem and divergence theorem, the general expression for conservation of linear momentum as applied to a control volume is:

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = \mathbf{f}. \quad (4.3)$$

Where $\mathbf{f} = \mathbf{f}_s + \mathbf{f}_b$ is the sum of the external surface forces \mathbf{f}_s and body forces \mathbf{f}_b acting on the control volume.

Surface forces

The forces acting on the control volume surface are due to pressure and viscous stresses which can be expressed in terms of the total stress tensor $\boldsymbol{\sigma}$ that in Cartesian coordinates is given by:

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{pmatrix} = \begin{pmatrix} -P & 0 & 0 \\ 0 & -P & 0 \\ 0 & 0 & -P \end{pmatrix} + \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{pmatrix} = -p\mathbf{I} + \boldsymbol{\tau}, \quad (4.4)$$

where \mathbf{I} is the identity tensor, p the pressure and $\boldsymbol{\tau}$ is the deviatoric of viscous stress tensor. The pressure is the negative part of the mean of the normal stresses and is given by:

$$p = -\frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}). \quad (4.5)$$

Hence the surface force acting on a differential surface element dS is:

$$\int_S \mathbf{f}_s dS = \int_A \boldsymbol{\sigma} \cdot \mathbf{n} dA = \int_V \nabla \cdot \boldsymbol{\sigma} dV \Rightarrow \mathbf{f}_s = \nabla \cdot \boldsymbol{\sigma} = -\nabla p + (\nabla \cdot \boldsymbol{\tau}). \quad (4.6)$$

Body forces

Body forces are forces per unit volume and the predominant ones are given below:

- Gravitational forces $\mathbf{f}_b = \rho \mathbf{g}$, due to the presence of a gravitational field
- Coriolis and centrifugal forces, respectively $\mathbf{f}_b = -2\rho(\boldsymbol{\omega} \times \mathbf{v}) - \rho(\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}))$, due to a rotating frame of reference

Hence introducing the expressions of surface and body forces in Eq.(4.3) the general conservative form of the momentum equation is obtained as:

$$\frac{\partial(\rho\mathbf{v})}{\partial t} + \nabla \cdot (\rho\mathbf{v}\mathbf{v}) = -\nabla p + (\nabla \cdot \boldsymbol{\tau}) + \mathbf{f}_b. \quad (4.7)$$

To proceed further the type of fluid should be specified in order to relate $\boldsymbol{\tau}$ with the other flow variables. For a Newtonian fluid the stress tensor is a linear function of the strain rate and is given by:

$$\boldsymbol{\tau} = \mu(\nabla\mathbf{v} + (\nabla\mathbf{v})^T) + \lambda(\nabla \cdot \mathbf{v})\mathbf{I}, \quad (4.8)$$

where μ is the molecular viscosity, λ the bulk viscosity coefficient usually set equal to $\lambda = \frac{2}{3}\mu$. Taking the divergence of Eq.(4.8) and substituting in Eq.(4.7) the final conservative form of the momentum equation for Newtonian fluids becomes also called Navier Stokes equation:

$$\frac{\partial(\rho\mathbf{v})}{\partial t} + \nabla \cdot (\rho\mathbf{v}\mathbf{v}) = \nabla \cdot (\mu\nabla\mathbf{v}) - \nabla p + \nabla \cdot (\mu(\nabla\mathbf{v}^T)) + \nabla(\lambda\nabla \cdot \mathbf{v}) + \mathbf{f}_b. \quad (4.9)$$

For incompressible flows the divergence of velocity vector is zero, $\nabla \cdot \mathbf{v} = 0$, and for constant molecular viscosity the momentum equation can be further simplified:

$$\frac{\partial(\rho\mathbf{v})}{\partial t} + \nabla \cdot (\rho\mathbf{v}\mathbf{v}) = -\nabla p + \mu\nabla^2\mathbf{v} + \mathbf{f}_b. \quad (4.10)$$

4.1.3 Conservation of Energy

The conservation of energy (the first law of thermodynamics) simply states that energy can be neither created nor destroyed during a process; it can only change from one form (mechanical, kinetic, chemical, etc.) into another. Consequently, the sum of all forms of energy in an isolated system remains constant. Considering a material volume MV of mass m , density ρ , and moving with a velocity \mathbf{v} the total energy E can be written as:

$$E = m(\hat{u} + \frac{1}{2}\mathbf{v} \cdot \mathbf{v}), \quad (4.11)$$

where \hat{u} is the internal energy per unit mass. The first law of thermodynamic states that the rate of change of the total energy of the material volume is equal to the rate of heat addition and work extraction through its boundaries:

$$\left(\frac{dE}{dt}\right)_{MV} = \dot{Q} - \dot{W}. \quad (4.12)$$

4.2 Turbulence modelling

Most of industrial applications involve turbulent flows. However a precise definition is somewhat difficult and all that can be done is a brief outline of some of its characteristics [18]. One characteristic is the irregularity, or randomness, of all turbulent flows. This makes a deterministic approach to problems including turbulence impossible; instead, statistical methods have to be relied on. Another important turbulence feature is its diffusivity that leads to rapid mixing, thereby increasing transfer rates of momentum, heat and mass through the flow domain. Turbulent flows always occur at a large Reynolds number, and often originate as the instability of laminar flows with increasing Reynolds numbers. Instabilities are related to the interaction of viscous terms and nonlinear inertia terms in the equations of motion. Turbulence is a 3-D phenomenon and there are no satisfactory 2-D approximation for determining fine details of turbulent flows; all turbulent flows are inherently dissipative and turbulence observe a cascade process whereby its kinetic transfer from larger eddies to smaller eddies and the latter dissipate into heat due to molecular viscosity. Turbulence is a continuum phenomenon governed by the equations of fluid mechanics. Even the smallest scales in any turbulent flow are much larger than any molecular length scale. Finally, turbulence is a flow feature, and not a fluid feature. It is possible to estimate the magnitude of the smallest scale through dimensional analysis. As stated above the cascade process involves a transfer of turbulent kinetic energy k (associated to fluctuating turbulent velocity) from larger eddies to smaller ones. The smaller eddies should be in a state where the rate of receiving energy from larger eddies is very nearly equal to the rate at which the smallest eddies dissipate the energy to heat [19]. Hence the motion at the smallest scales should depend only upon the rate at which the larger eddies supply energy, $\epsilon = -\frac{dk}{dt}$ and the kinematic viscosity ν . Having established appropriate dimensional quantities for ϵ and ν one can derive the *Kolmogorov* scales of length, time and velocity

$$\eta = \left(\frac{\nu^3}{\epsilon}\right)^{\frac{1}{4}}, \quad \tau = \left(\frac{\nu}{\epsilon}\right)^{\frac{1}{2}}, \quad v = (\nu\epsilon)^{\frac{1}{4}}. \quad (4.13)$$

With dimensional analysis the dissipation rate ϵ could be related with k through:

$$\epsilon \sim \frac{k^{\frac{3}{2}}}{l}, \quad (4.14)$$

where l is the integral length scale of the largest eddies. Hence the ratio

$$\frac{l}{\eta} \sim Re_t^{\frac{3}{4}}, \quad (4.15)$$

with Re_t being the turbulence Reynolds number based on l and k . Thus, the energy cascade involves a number of scales proportional to N :

$$N = Re_t^{\frac{9}{4}}. \quad (4.16)$$

It is now clear that in order to ensure that all the features of turbulence are predicted correctly, a large computational domain and a very dense grid are requested. This is a *DNS* (direct numerical simulation) approach and it is not affordable for industrial applications because of the need to obtain results within a reasonable time and because of the great request of computational resources. Hence a mathematical model is required to predict turbulent flow properties but modelling turbulence involves statistical studies of the equations of fluid flow and always leads to the closure problem: more unknowns than equations. In order to make the number of equations equal to the number of unknowns, assumptions are imperative. Usually there are two approaches: filtering in space or averaging in time. The first approach called *LES* (large eddy simulation) consists on applying a spatial filter to Navier Stokes equations with only the length scales smaller than the size of the filter modelled. Nevertheless, nowadays time averaging is still the most common turbulence model approach in industrial applications and all turbulent fluctuations need to be modelled. The key approach is to decompose the flow variables into a time-mean value component and a fluctuating one, substituting in the original equations, and time-averaging the obtained equations. Expressing the instantaneous velocity as the sum of a mean and a fluctuating part so that:

$$\mathbf{v}(\mathbf{x}, t) = \bar{\mathbf{v}}(\mathbf{x}) + \mathbf{v}'(\mathbf{x}, t). \quad (4.17)$$

The time-averaging properties lead to the following expression for the incompressible *RANS* continuity, momentum, energy equations:

$$\nabla \cdot (\rho \bar{\mathbf{v}}) = 0, \quad (4.18)$$

$$\frac{\partial \rho \bar{\mathbf{v}}}{\partial t} + \nabla \cdot \{\rho \bar{\mathbf{v}} \bar{\mathbf{v}}\} = -\nabla \bar{p} + \nabla \cdot (\bar{\boldsymbol{\tau}} - \rho \overline{\mathbf{v}' \mathbf{v}'}) + \rho \mathbf{g}, \quad (4.19)$$

$$\frac{\partial}{\partial t} (\rho c_p \bar{T}) + \nabla \cdot (\rho c_p \bar{\mathbf{v}} \bar{T}) = \nabla \cdot (k \nabla \bar{T} - \rho c_p \overline{\mathbf{v}' T'}) + \bar{S}^T. \quad (4.20)$$

Keeping the unsteady term $\frac{\partial \rho \bar{\mathbf{v}}}{\partial t}$ in the momentum equation usually brings to the definitions of *URANS* (unsteady Reynolds averaged Navier-Stokes), but attention should be maintained for those turbulent flows where there is no clear distinction

between timescale characteristic of slow variations of the mean flow and that related to turbulent fluctuations. Indeed, the approximation

$$\overline{\frac{\partial \mathbf{v}(\mathbf{x}, t)}{\partial t}} \approx \frac{\partial \bar{\mathbf{v}}(\mathbf{x})}{\partial t}, \quad (4.21)$$

is true if $|\mathbf{v}'| \ll |\bar{\mathbf{v}}|$. This is always questionable, however using time averaging in this manner is useful for analysis especially for time marching numerical methods implemented for solving fluid dynamics problems but a degree of caution must be exercised when fluctuations are not too small. Comparing Eq.(4.19) with Eq.(4.7) and the Incompressible Energy equation with Eq.(4.20), one can note the appearance of new terms on the right-hand-side. These terms are called *Reynolds Stresses Tensor* and *turbulent heat fluxes*. So what Reynolds-averaging does is to introduce 9 new variables and to solve the *RANS* (Reynolds averaged Navier-Stokes) equations, but additional equations are required.

Here comes into play the *Boussinesq Hypothesis* which makes an analogy with Newtonian fluids by assuming that the *Reynolds stresses* are a linear function of the *mean velocity gradients*

$$-\overline{\rho \mathbf{v}' \mathbf{v}'} = \mu_T [\nabla \mathbf{v} + (\nabla \mathbf{v})^T] - \frac{2}{3} \rho k \mathbf{I}. \quad (4.22)$$

This assumption reduces the number of unknown from 6 to 2: the *turbulent eddy viscosity* μ_T and the *turbulent kinetic energy* k .

For incompressible flows, the equations can be rearranged by defining a *turbulent pressure* p [13]:

$$p \leftarrow p + \frac{2}{3} \rho k. \quad (4.23)$$

In this manner, the only unknown that remains to compute is the *turbulent eddy viscosity* μ_T . The great variety of turbulence models derive from different ways of evaluating μ_t . In a similar way, the turbulent thermal fluxes are calculated in analogy with Fourier's law such that

$$-\rho c_p \overline{\mathbf{v}' T'} = \alpha_t \nabla T, \quad (4.24)$$

where α_t is the turbulent thermal diffusivity.

In the following section are presented the principle and the capabilities of the *Shear Stress Transport (SST) k - ω model* that is the one employed in this work. However for a better understanding also the $k - \epsilon$ and the $k - \omega$ models are briefly explained.

4.3 Shear Stress Transport $k - \omega$ Model

The $k-\omega$ family of linear eddy viscosity models seems to be by far the most widely used ones. Before describing the *SST* $k - \omega$ model it is necessary to briefly introduce *standard* $k - \epsilon$ and $k - \omega$ models, since *SST* is a combination of these two approach. Both methods belong to the *two-equations* family of turbulence models. This class involves the resolution of two additional partial differential equations in order to locally compute the *turbulent eddy viscosity* μ_T and the *turbulent thermal diffusivity* α_t .

$k - \epsilon$

Like other models based on the *Boussinesq Hypothesis*, the $k - \epsilon$ model is based on the following expressions for *turbulent eddy viscosity* μ_t and for turbulent thermal diffusivity α_t :

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon}, \quad \alpha_t = c_p \frac{\mu_t}{Pr}, \quad (4.25)$$

where C_μ is a calibration constant, k is the *turbulent kinetic energy* and ϵ is the *rate of dissipation of turbulent kinetic energy per unit mass due to viscous stresses*. Solving the following transport equations for k and ϵ a local value of μ_t can be computed:

$$\frac{\partial}{\partial t}(\rho k) + \nabla \cdot (\rho \mathbf{v} k) = \nabla \cdot \left(\left(\mu + \frac{\mu_T}{\sigma_k} \right) \nabla k \right) + S_k, \quad (4.26)$$

$$\frac{\partial}{\partial t}(\rho \epsilon) + \nabla \cdot (\rho \mathbf{v} \epsilon) = \nabla \cdot \left(\left(\mu + \frac{\mu_T}{\sigma_\epsilon} \right) \nabla \epsilon \right) + S_\epsilon, \quad (4.27)$$

It must be kept in mind that the construction of this model is based on two important assumptions:

- *fully turbulent flow*
- *negligible molecular viscosity effects*

that establish the limits of this approach:

- *validity only for high Reynolds*
- *inability to reach the wall*

To account for this lack the so-called *low Reynolds* $k - \epsilon$ model have been developed. These models use damping functions to damp the turbulent viscosity while getting close to the wall.

$k - \omega$

In this model the equation for ϵ is substituted by an equation for ω , where ω is called *specific turbulent dissipation* and represents the rate at which the turbulent kinetic energy is converted into thermal energy per unit time and unit volume.

$$\omega = \frac{\epsilon}{C_\mu k}. \quad (4.28)$$

The *turbulent eddy viscosity* and *turbulent thermal diffusivity* are then given by:

$$\mu_T = \rho \frac{k}{\omega}, \quad k_T = \frac{\mu_T}{Pr_T}. \quad (4.29)$$

The two additional equations are:

$$\frac{\partial}{\partial t}(\rho k) + \nabla \cdot (\rho \mathbf{v} k) = \nabla \cdot \left(\left(\mu + \frac{\mu_T}{\sigma_k} \right) \nabla k \right) + S_k, \quad (4.30)$$

$$\frac{\partial}{\partial t}(\rho \omega) + \nabla \cdot (\rho \mathbf{v} \omega) = \nabla \cdot \left(\left(\mu + \frac{\mu_T}{\sigma_\omega} \right) \nabla \omega \right) + S_\omega. \quad (4.31)$$

This new equation has three advantages [13]:

- *it is easier to integrate;*
- *it is integrable also in the sub-layer without using damping functions;*
- *it is capable to deal with weak adverse pressure gradients.*

As pointed out by its inventor, the $k - \omega$ model is accurate for both free shear flows and wall-bounded (attached boundary layer and mildly separation) [19]. But, unfortunately, this model has a strong dependence on the free stream values.

Looking at capabilities and flaws of the two models, they seems to be "complementary". The $k - \epsilon$ model, thanks to its insensitivity to the free stream, predicts with more accuracy away from the wall, while the $k - \omega$ behaves better in the boundary layer and with weak adverse pressure gradients.

These considerations have led to the development of the *Baseline (BSL) $k - \omega$* model, that uses a *blending function* to switch from the $k - \omega$ and a rearrangement of $k - \epsilon$ in terms of ω .

The *Shear Stress Transport $k - \omega$* model represents a further improvement to the *Baseline*, by limiting the shear stress in adverse pressure gradient flows. Menter, the developer of these two methods, writes [17]: "*It (BSL) has a performance similar to the Wilcox model, but avoids that model's strong freestream sensitivity. The second*

model (SST) results from a modification to the definition of the eddy-viscosity in the BSL model, which accounts for the effect of the transport of the principal turbulent shear stress. The new model is called shear-stress transport-model and leads to major improvements in the prediction of adverse pressure gradient flows.”

4.4 Near the wall treatment

On every solid surface, due to the fluid viscosity, a boundary layer develops. This layer of fluid can be divided in three regions:

- *viscous sub-layer* ($0 < y^+ < 5$), where the effect of viscosity dominates;
- *buffer sub-layer* ($5 < y^+ < 30$), where viscous and inertial effects are equal;
- *inertial (log-law) sub-layer* ($30 < y^+ < 500$), where the effect of inertia dominates.

These three sub-layers can be identified by the value of y^+ that is the adimensionalized normal distance (d_{\perp}) from the wall:

$$y^+ = \frac{d_{\perp} u_{\tau}}{\nu}, \quad (4.32)$$

where $u_{\tau} = \sqrt{\tau_w/\rho}$ is the velocity scale.

This subdivision of the boundary layer is schematized in *Figure 4.1*.

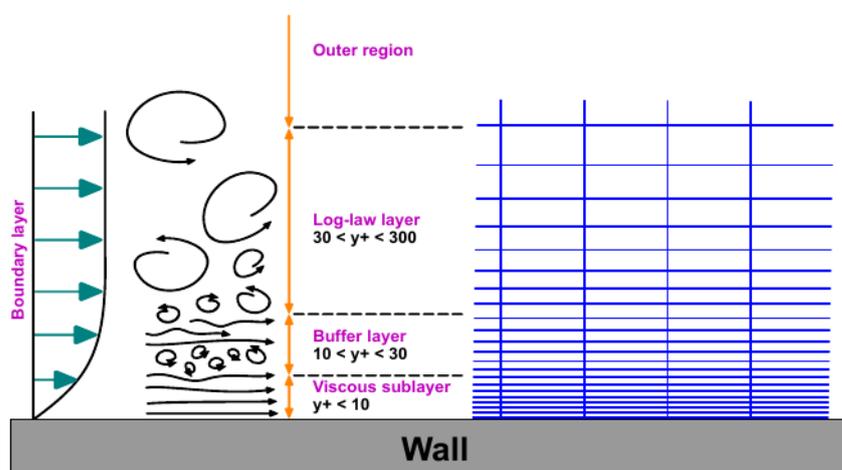


Figure 4.1: *Boundary layer subdivision and correspondent y^+ ranges (courtesy of Wolf-Dynamics srl [6])*

Turbulence models avoid the *buffer sub-layer*, because the high turbulent production, by placing the first cell center in the *viscous sub-layer* or in the *inertial sub-layer*.

The first option leads to accurate prediction of the boundary layer, but requires a very fine discretization near the wall, usually leading to unaffordable costs.

The second, combined by the definition an appropriate wall-value to each new variable introduced, significantly reduces computational costs while giving a good accuracy. This velocity profile is called *wall function* and its action is schematized in *Figure 4.2*.

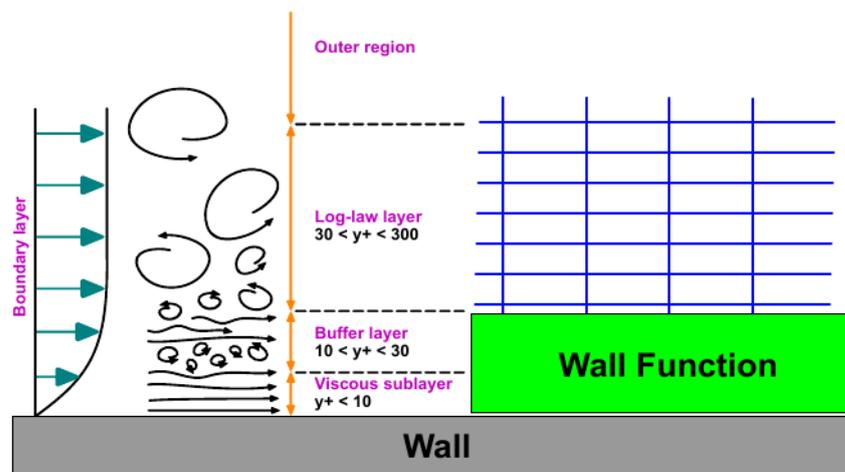


Figure 4.2: Representation of the wall function approach. (courtesy of Wolf Dynamics srl [6])

Chapter 5

RANS Simulations with MRF Method

The aim of the first simulations was to characterize the fluid motion due to the fan rotation inside the chamber. Several cases were run in order to investigate the influence of the fan rotational speed on the fluid field. Nevertheless two experimental setups of ChAMBRé have been reproduced. In the first one the only element inside the chamber is the fan and in the second case the rectangular plate with petri capsules is installed inside the chamber, above the fan. For a clear understanding these two cases will be respectively called: *noPlate* case and *Plate* case. In this part of the work we needed relatively fast convergence but without losing too much accuracy. Therefore the chosen configuration for all the cases is here shortly listed:

- *Coarse Mesh*
- RANS Reynolds Averaged Navier Stokes equations;
- *SST* $k - \omega$ turbulence model;
- MRF multiple reference frame;

The following sections briefly describe the mesh generation process, the MRF method and the results obtained.

5.1 *Coarse Mesh* generation

The mesh generation has been carried out basically with two OPENFOAM[®] utilities: *blockMesh* and *snappyHexMesh* that are controlled respectively by *blockMesh-*

Dict and snappyHexMeshDict files. With *blockMesh* the background mesh is obtained by decomposing an user defined domain into a set of hexahedral blocks. The domain chosen is a parallelepiped that must contain the whole geometry. An OPENFOAM[®] utility called *surfaceCheck* can suggest the proper coordinates of the edges of the parallelepiped and the spatial discretization can be defined by the user. Once the background mesh is obtained the *snappyHexMesh* utility generates the actual mesh containing hexahedra and split-hexahedra from the surface geometries. The first step in the meshing process is the cell splitting according to specification defined. Here the user can select several levels and types of refinement for different surfaces, volumes and also edges with the dedicated utility *surfaceFeatureExtract*. Briefly the next steps remove and snap the cells then add layers on walls if needed. During the process the mesh quality is controlled by the entries in the sub-dictionary *meshQualityControls*. After the meshing process is completed, the OPENFOAM[®] utility *checkMesh* will output the features of the mesh. The most important parameters for checking the quality of the mesh are:

- *Mesh orthogonality*: This parameter is related to angular deviation of the cell face normal vector from the vector connecting two consecutive cell centres. Usually this parameter is kept below 75.
- *Mesh skewness*: Skewness is the deviation of the vector that connects two cell centres from the face centres. Usually this parameter should be kept below 8.
- *Aspect ratio*: Aspect ratio AR is the ratio between the longest side Δx and the shortest side Δy of the cell. Large AR are good if gradients in the largest direction are small.

These three mesh features are schematized in Fig.5.1.

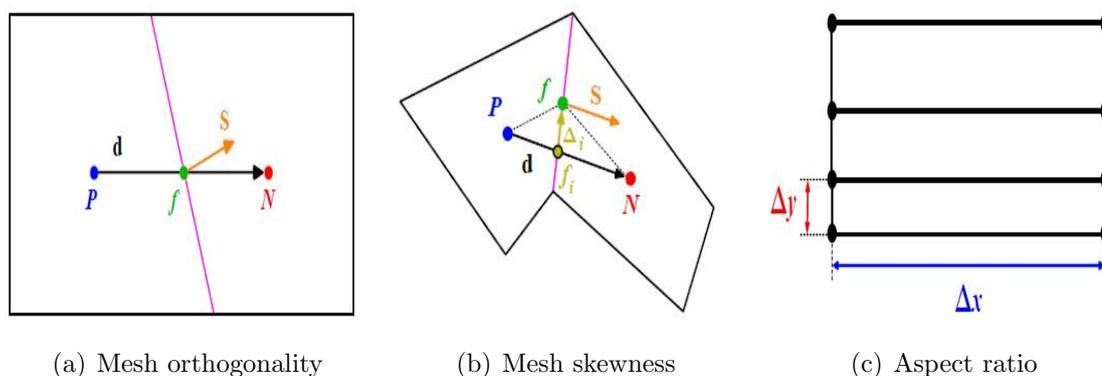


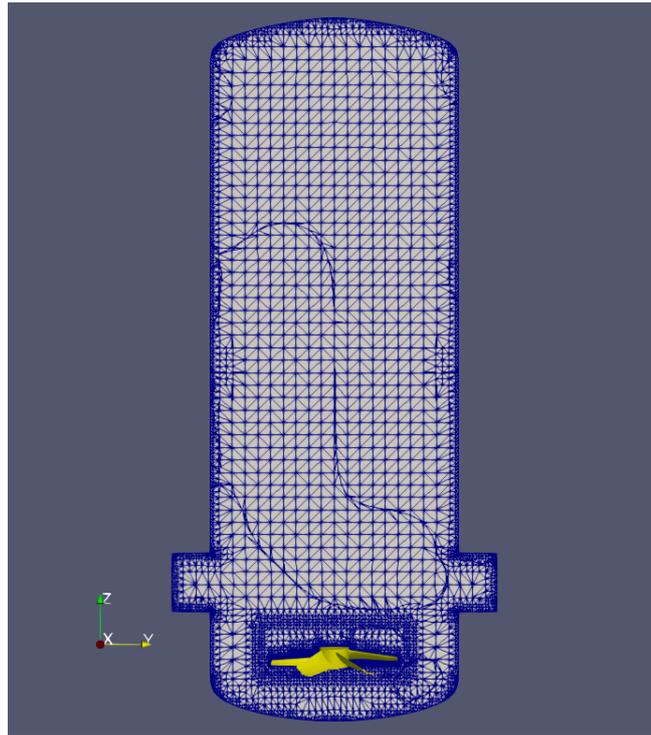
Figure 5.1: *Main mesh quality parameters*

Recalling the two experimental setups: with and without plate, it is clear that two meshes have been made but the generating criteria were the same in both cases. In Tab.5.1 are listed the main features of the two meshes.

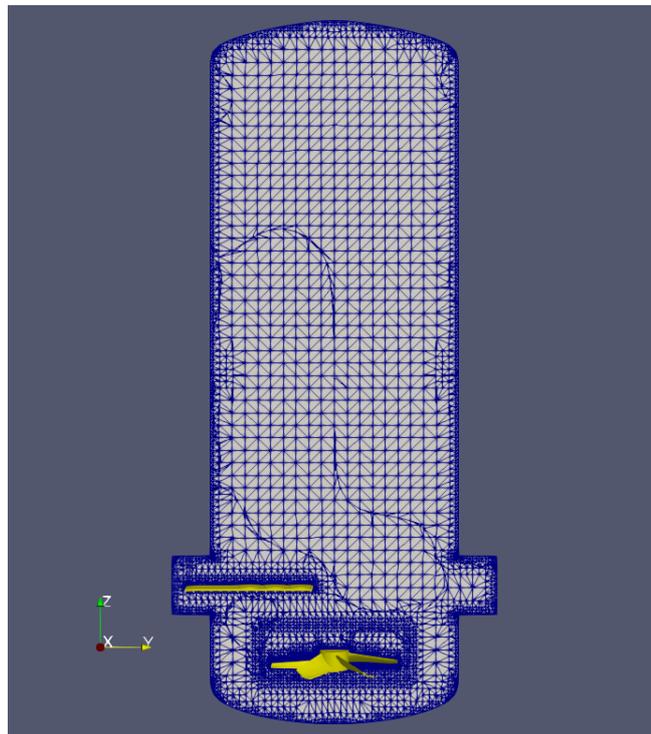
Table 5.1: *Mesh features*

	Coarse Mesh	Coarse Mesh with Plate
Cells	1,472,285	1,472,144
HexaHedra	828,156	840,438
Prisms	86,154	85,123
Pyramids	0	0
Tetrahedra	95	99
Polyhedra	549,465	538,396
Non-orthogonality	Max= 75.00 Average= 16.02	Max= 75.00 Average= 16.02
Max skewness	3.998	3.999
Max <i>AR</i>	36.28	36.28
Number of layers	0 chamber, 3 fan	0 chamber, 3 fan, 0 plate

In this part of the study in order to avoid high computational efforts we kept the number of cells relatively low, generating coarse meshes. For this reason the *snappyHexMeshDict* has been set up with low refinement levels on the chamber while the fan has been highly refined and three thin layers have been added. This choice is reasonable because the flow field is entirely generated by the fan that provides the mixing inside the chamber. Furthermore resolving with high accuracy regions near chamber wall and inside the flanges will be extremely time consuming. In the case *Coarse mesh with Plate* we selected a medium refinement for plate and petri capsules because these components will be fundamental for the particles simulations. In Fig.5.2 the two meshes are visualized in a vertical plane section using the OPENFOAM® utility *paraView*. As can be seen in Fig.5.2 a well refined cylindrical region that contains the fan has been created with the purpose of using the multiple reference frame method that is going to be introduced in the next section.



(a) *Coarse Mesh*



(b) *Coarse Mesh with Plate*

Figure 5.2: *Mesh visualization with paraView*

5.2 Multiple Reference Frame Method

The Multiple Reference Frame or MRF consists in a method for modelling cases with multiple moving zones. A variety of problems can be solved with this approach such as impeller in mixing tanks, axial fans or cooling ducts. The method is not suggested in cases where stator-rotor interaction are really important. The case study seems to be suitable for the MRF because essentially ChAMBRé is a vessel with a rotating fan at the bottom. MRF method is also useful in computing a flow field that can be set as initial condition of a transient sliding mesh calculation [7]. The MRF implementation in the computational domain starts from dividing the domain in inertial and rotating subdomain. To obtain the absolute velocity, the relative velocity in the moving reference frame has to be added to the velocity in the rotational zone using the following equation [8]:

$$\mathbf{v} = \mathbf{v}_r + \bar{\omega} \times \mathbf{v} + \mathbf{v}_t \quad (5.1)$$

Where \mathbf{v} is the velocity in the absolute inertial reference frame, \mathbf{v}_r is defined as the velocity in the relative frame and \mathbf{v}_t is the translational velocity of non-inertial reference frame. In the inertial frame the governing equations is the Navier Stokes equation Eq.(4.10) for incompressible and steady flows.

$$\begin{aligned} \nabla \cdot \mathbf{v} &= 0 \\ \nabla \cdot (\rho \mathbf{v} \mathbf{v}) &= -\nabla p + \mu \nabla^2 \mathbf{v} \end{aligned} \quad (5.2)$$

With the previous hypothesis, in the rotating frame by applying the Eq.(5.1) in the momentum equation and solving in terms of relative velocity result:

$$\begin{aligned} \nabla \cdot \mathbf{v}_r &= 0 \\ \nabla \cdot (\rho \mathbf{v}_r \mathbf{v}_r) + \rho(2\bar{\omega} \times \mathbf{v}_r) + \bar{\omega} \times \bar{\omega} \times \mathbf{r} &= -\nabla p + \mu \nabla^2 \mathbf{v} \end{aligned} \quad (5.3)$$

Where the second term is the additional Coriolis and the third term is defined as centripetal acceleration.

In the case study a cylindrical domain including and surrounding the fan is the rotating frame while other parts are included in the inertial subdomain. In OPENFOAM® during the mesh generation the rotating frame is associated to a user defined *cell-Zone* that will rotate around a selected axis with a given rotational speed.

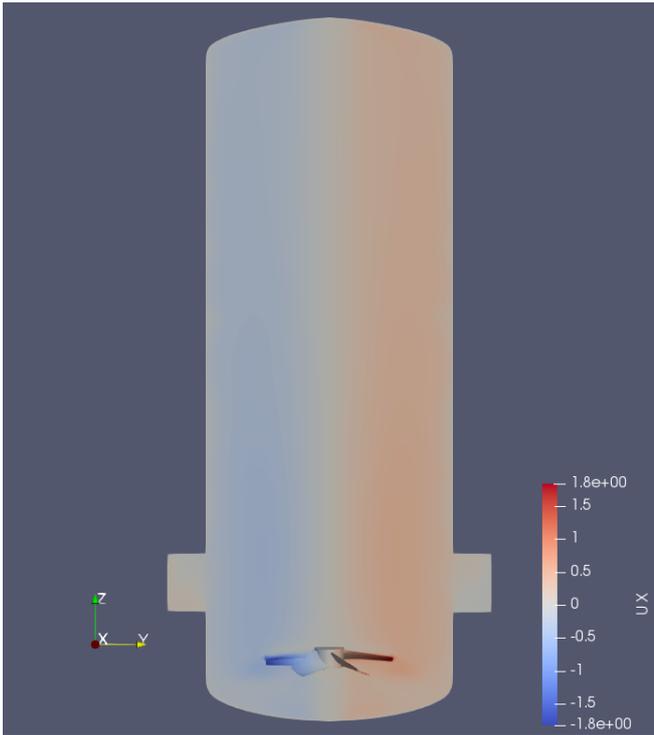
5.3 Results

Every simulation has been run with the same OPENFOAM[®] solver: *simpleFoam*, a steady-state solver for incompressible, turbulent flow. The solver settings in file `fvSolution` `fvSchemes` have been kept constant. The only exceptions are the relaxation factors that started from low values in order to avoid divergence of the solution. Successively after an adequate number of iterations the factors have been increased to standard values. The postprocessing of the results is made with *paraView* and follows two basic ideas. For single cases the contours of velocity are visualized on a section plane that cut in the half the simulations domain. For the parametric analysis we plotted components of velocity along lines at three different heights. From now on the velocity component with the direction of x axis will be called U_x and the other components will be U_y and U_z .

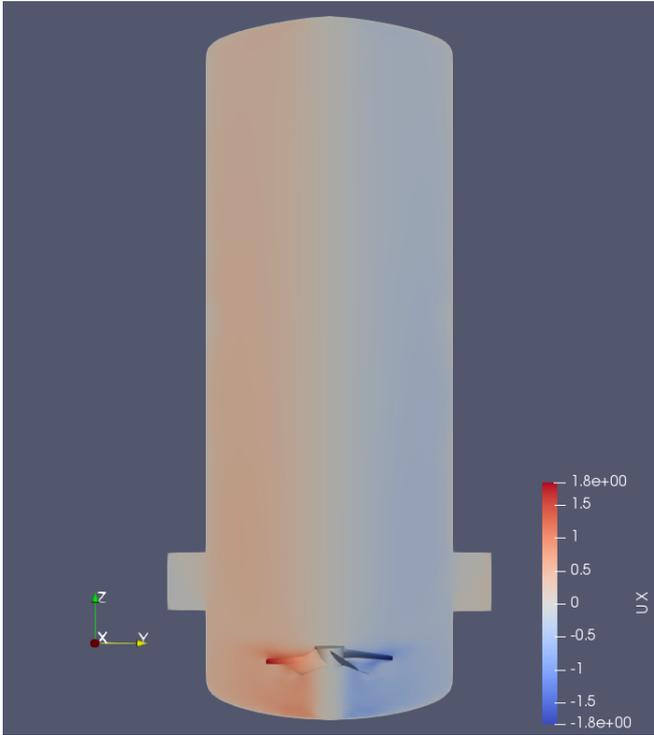
5.3.1 *noPlate* simulations

Clockwise and *Counterclockwise* comparison

The analysis of the results starts from the *noPlate* case so the actual setup presents only the fan inside ChAMBRé. First of all we compared two cases at the same frequency of rotation of the fan $f = 1Hz$ but with opposite rotational direction. The *Clockwise* and the *Counterclockwise* cases present similar flows. In Fig.(5.9) the U_x field is coloured with red for positive values, with blue for the negative ones. In this precise section that is normal to x the U_x component can be seen as the tangential velocity in a cylindrical coordinates system with the chamber axis as reference axis. Briefly U_x indicates the rotational motion of the air in the whole chamber caused by the fan. As can be seen in Fig.(5.9a) the direction of rotation is *Clockwise* while in Fig.(5.9b) it is *Counterclockwise*. Visualizing U_y on the same section will not give any other important information on the flow from the moment that the y component is near zero on this surface. A more interesting component is U_z because it can be argued that the vertical velocity will be fundamental during a mixing process in a vessel. From a look at Fig.(5.4) it is clear that for both cases there are slight differences but the tendency of the flow is the same.

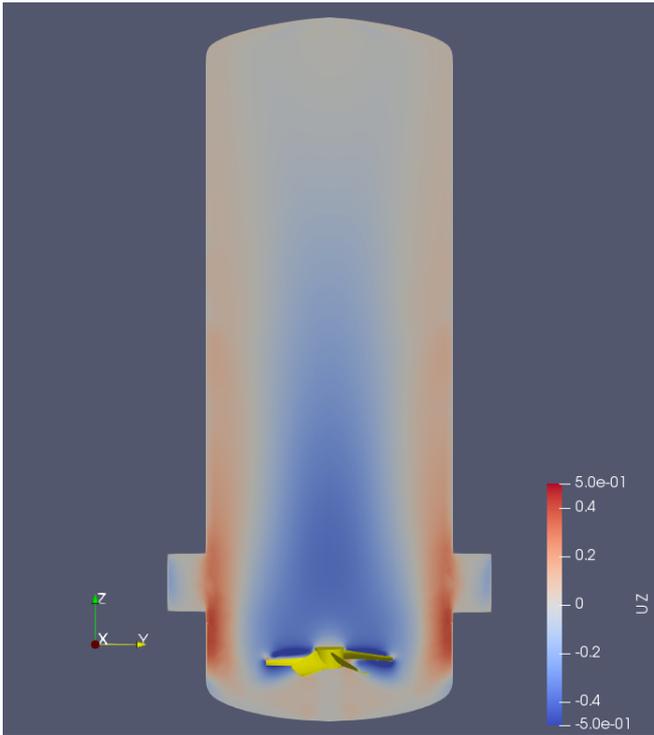


(a) *Clockwise*

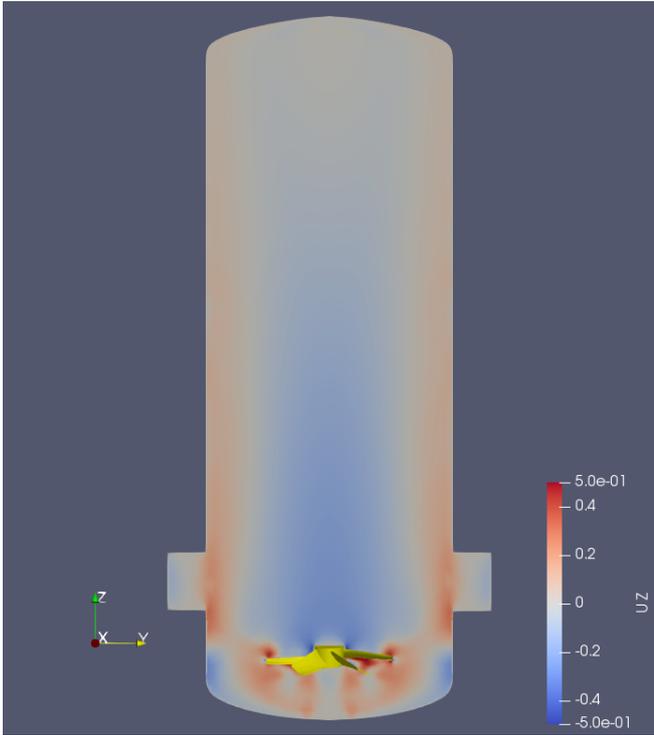


(b) *Counterclockwise*

Figure 5.3: *Contour of U_x on vertical section x – normal*



(a) *Clockwise*



(b) *Counterclockwise*

Figure 5.4: *Contour of U_z on vertical section $x - normal$*

A global recirculation is happening inside ChAMBRé: the rotating flow ascends near the walls and it is sucked at the centre, forming a big column of air. This similarity between *Clockwise* and *Counterclockwise* fields can seem strange and has been deeply studied starting from the element that generates this behavior: the fan. A single blade is greatly twisted and tapered so this shape will push the air upwards only on the tips of the blades. This spanwise effect is stronger than the effect that the 2D shape of the profile can induce to the flow. For this reason the solution is not strongly dependent on the direction of the rotation. A consultation with the customer service of the fan producer *Elektrovent* confirmed this hypothesis. Concluding it should be noticed that the flux is generally axisymmetric in the main volume of ChAMBRé while this feature is not kept in the flanges because these aren't symmetrically disposed in the geometry.

Results of the Parametric study

After knowing the main characteristics of the flow we started a parametric study with the following simulations:

- *Clockwise* with $f = [1, 2, 3, 4, 5, 7]Hz$
- *Counterclockwise* with $f = [1, 2, 3, 4, 5, 7]Hz$

Totally twelve simulations were carried out and the same analysis previously described has been done for every single case. It is really important to note that the resulting fields were really similar for both directions of rotation as in the case of for $f = 1Hz$. The increasing of the rotational speed of the fan doesn't change the basic feature of the flow. It's still present the same recirculating motion that affects the whole chamber and only slight differences are noticed in the directions of velocity. It turns out that a variation of f essentially changes the magnitude of velocity $|U|$. This dependance was predictable considering that during this parametric study we are increasing the speed of the fan, the component that generates the motion. The results of these simulations are collected with plots which show velocity profiles along a line parallel to y axis. This profile is represented at three heights measured from the bottom wall of the chamber:

- *Height* = 0.4[m] just above the fan
- *Height* = 1.3[m] in the middle of ChAMBRé
- *Height* = 2.3[m] at 0.7[m] below the top wall

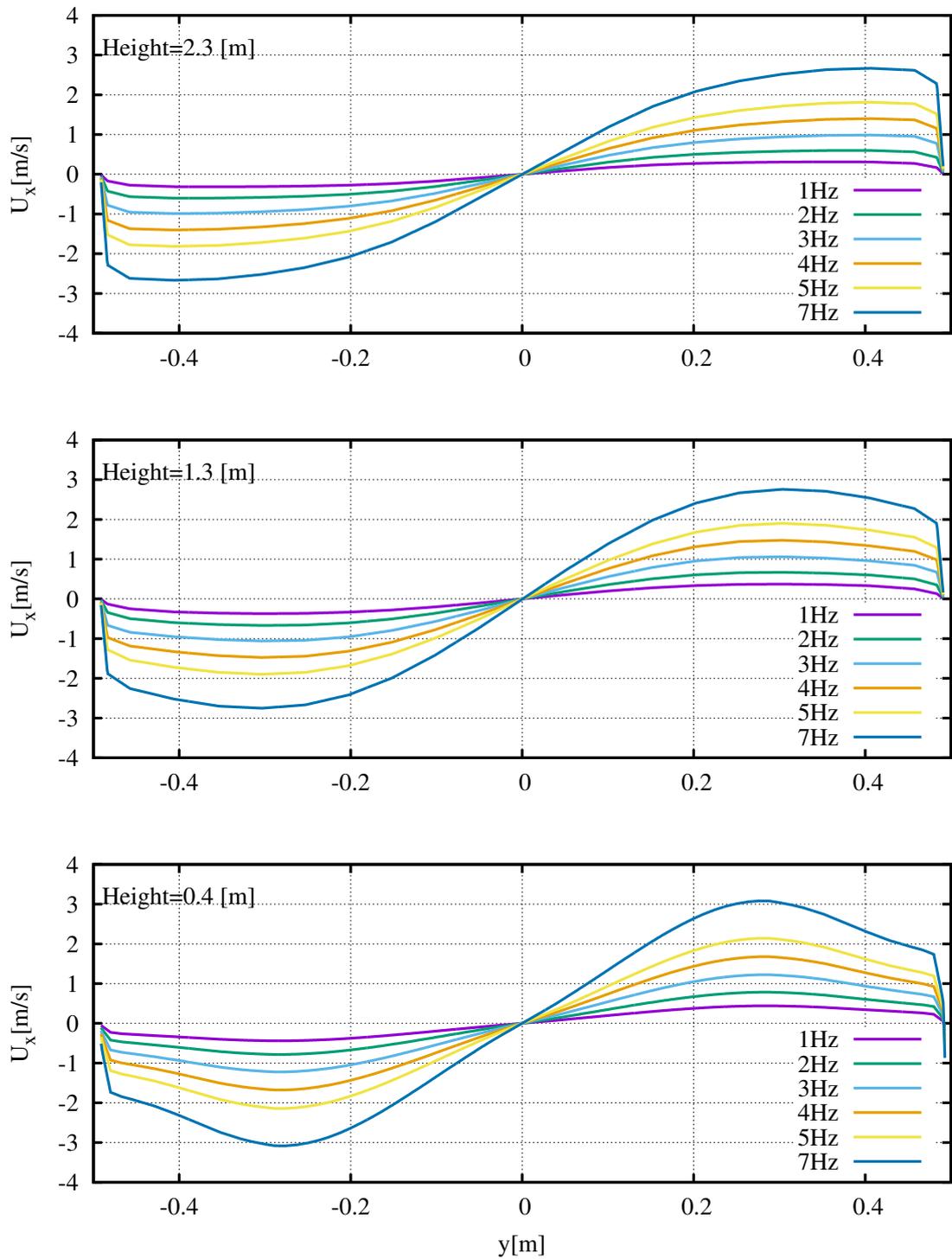
For the *Clockwise* case:

- In Fig.(5.5), from the profiles of U_x can be clearly seen that the rotational motion of the fluid is conserved at different heights and the values of the component grow with the frequency.
- In Fig.(5.6) first should be noticed that the profiles of U_z takes a lower values range than profiles of U_x , almost the half. Generally the recirculation is present with an upward motion near walls and a descending column of air at the center. Furthermore U_z strongly decreases with the growing of the height. At $Height = 2.3[m]$ values are really close to zero so the small vertical velocities generated at bottom by the fan are completely dissipated near the top dome. For mixing purpose this feature is really positive because the risk of pushing the mixture into the the top wall is strongly reduced.

As previously stated for $f = 1Hz$ the rotational direction should not strongly influence the flow field. In fact the *Counterclockwise* parametric study led us to the same conclusions enforcing our hypothesis.

- In Fig.(5.7) U_x is plotted and has the features of the *Clockwise* case with the obvious exception for the direction of rotation.
- In Fig.(5.8) U_z is plotted. When comparing these profiles with *Clockwise* case, the tendency are similar but the values of U_z in this case are lower.

From a further analysis of the fan blade profiles, supported by these results, we concluded that the *Clockwise* direction of rotation will be the proper direction for this machine in the experimental setup.

Figure 5.5: Clockwise U_x plots along y at three Heights with f as parameter

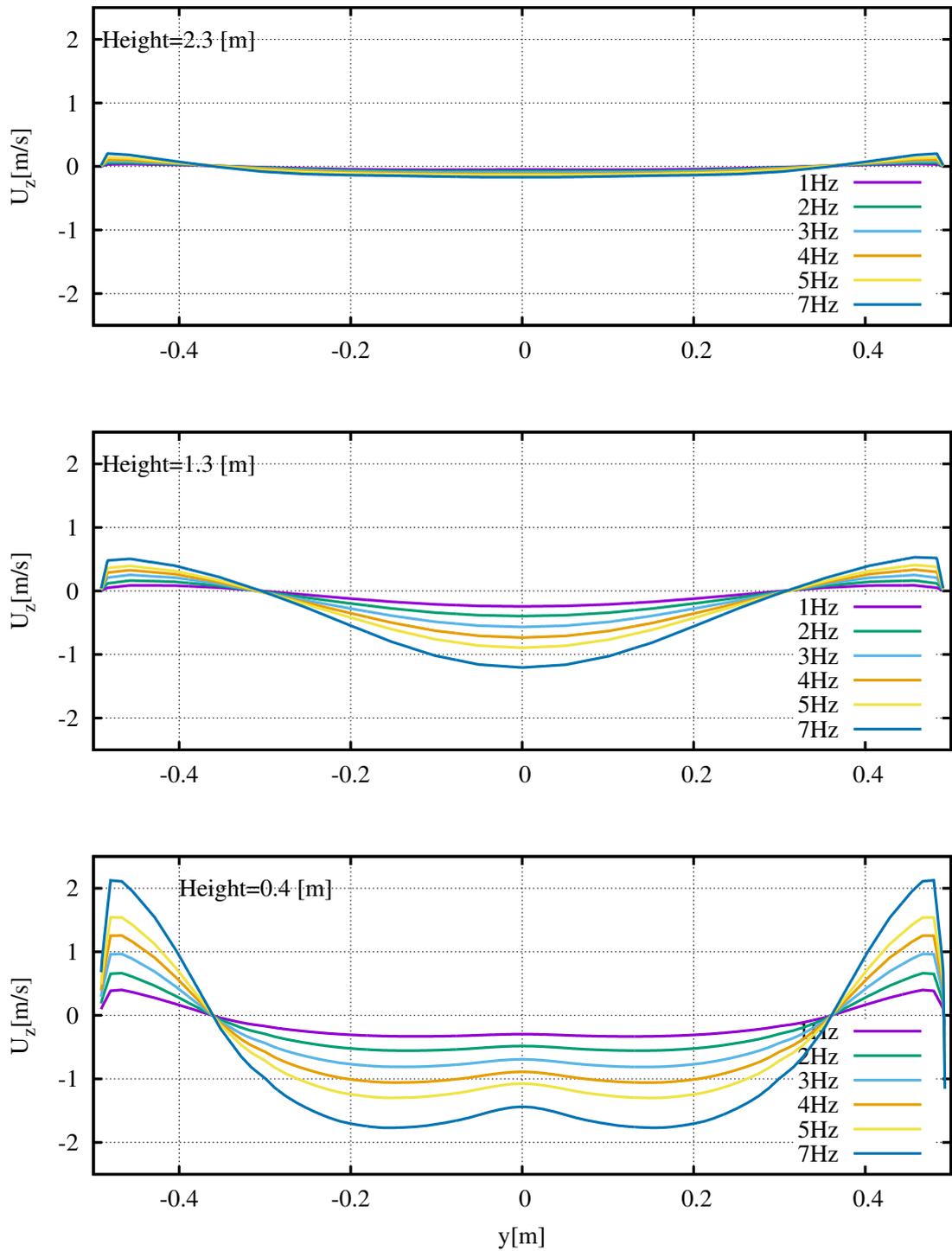


Figure 5.6: Clockwise U_z plots along y at three Heights with f as parameter

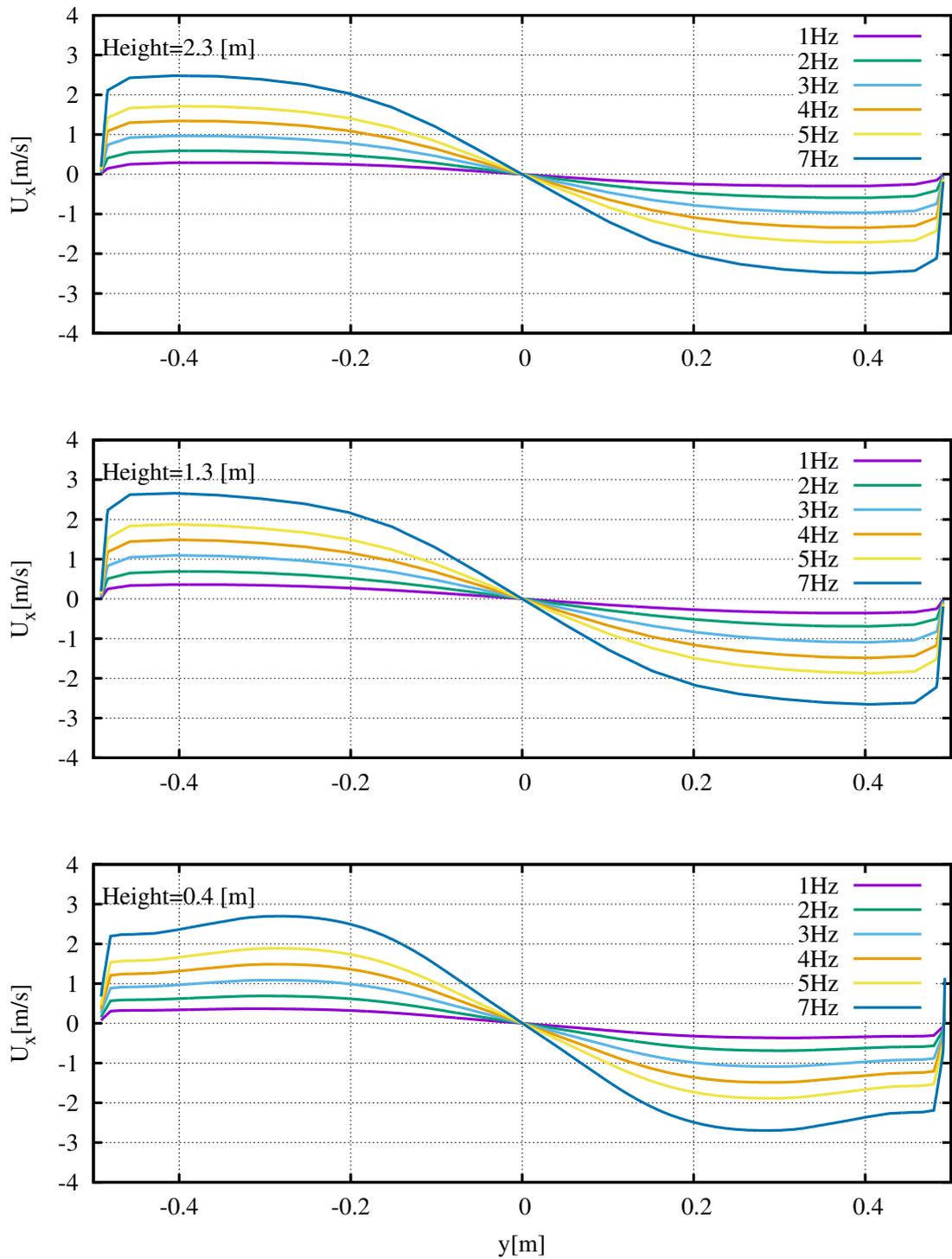


Figure 5.7: Counterclockwise U_x plots along y at three Heights with f as parameter

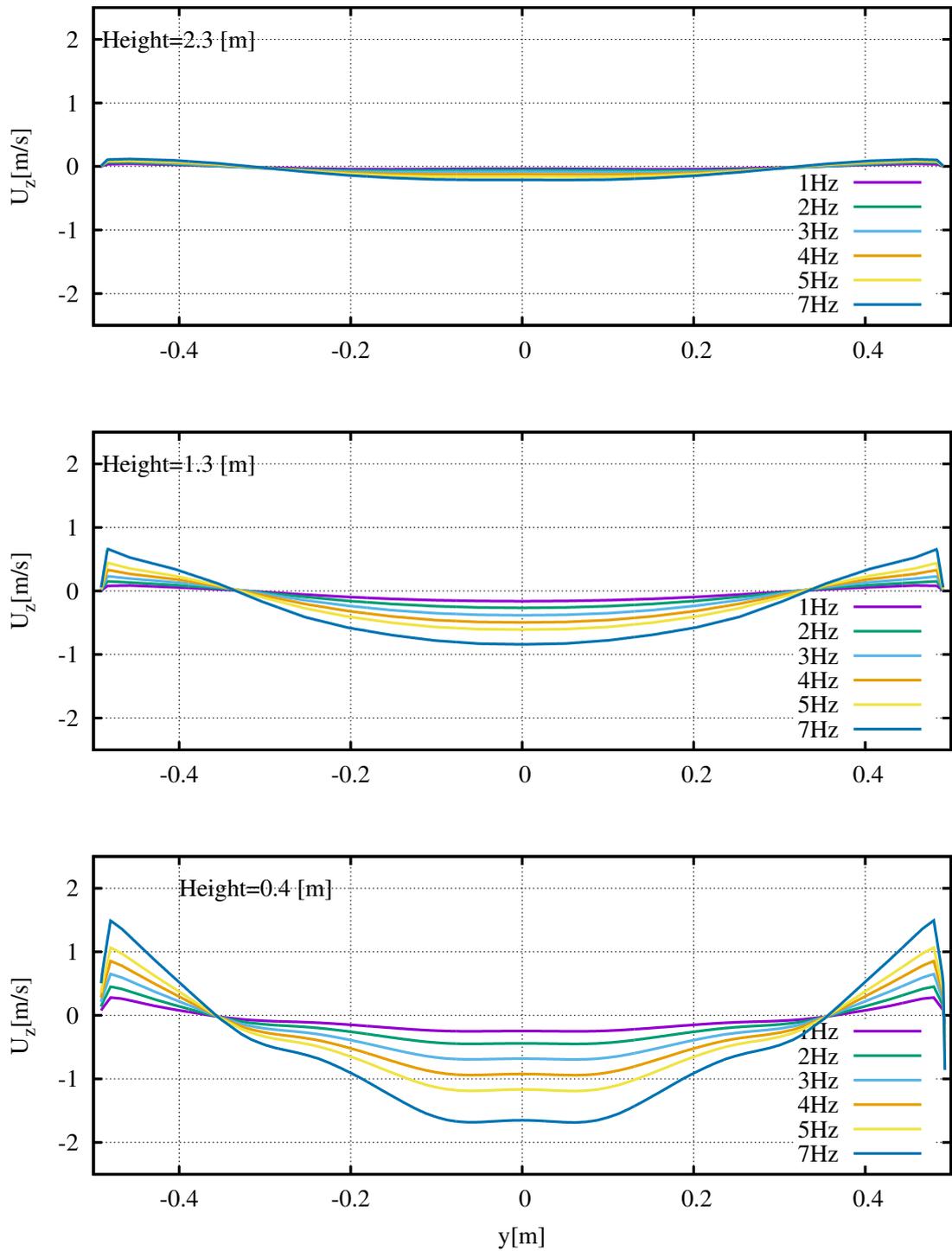
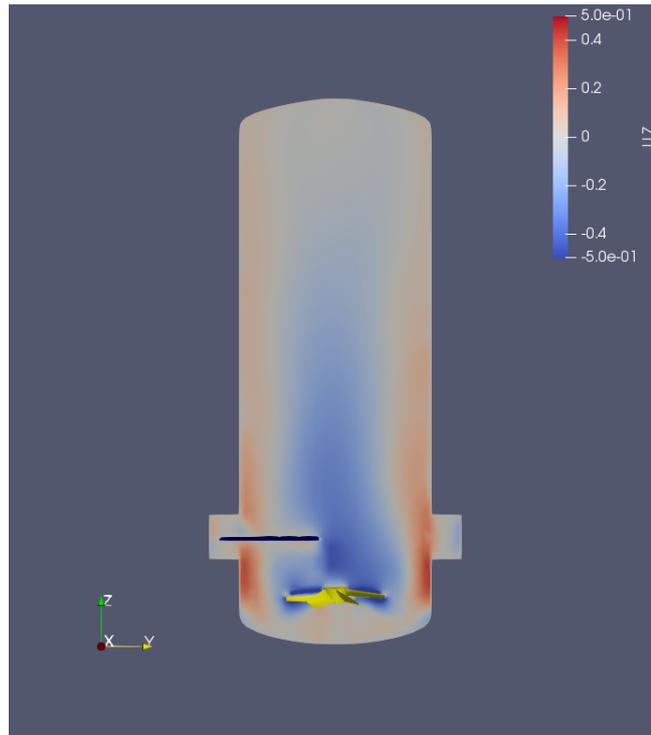


Figure 5.8: Counterclockwise U_z plots along y at three Heights with f as parameter

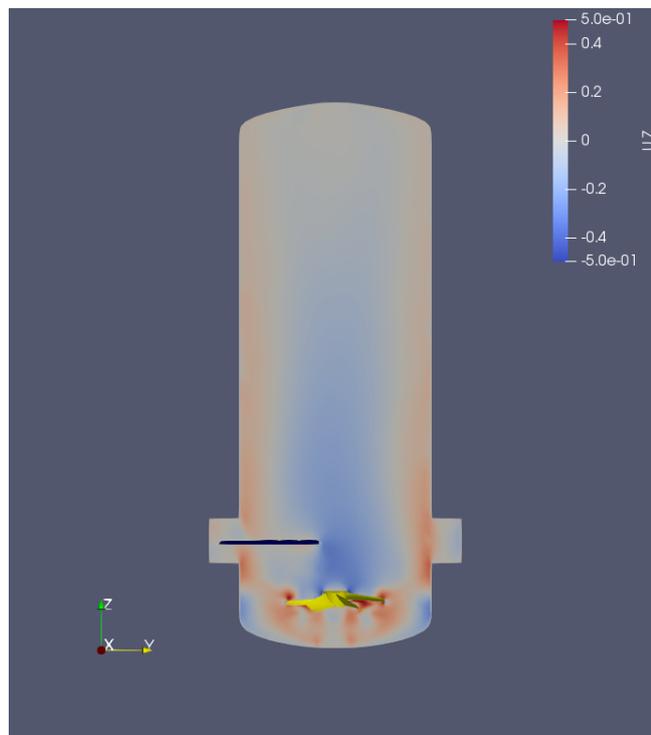
5.3.2 *Plate simulations*

Clockwise and Counterclockwise comparison

The analysis of the results continues with the *Plate* case. The rectangular plate with petri capsules is installed inside ChAMBRé, above the fan. Again we started from the comparison of two cases at the same frequency of rotation of the fan $f = 1Hz$ but with opposite rotational directions. The U_x field is not represented because the result is really close to the field from the *noPlate* case. It has to be noticed that the rectangular plate has a thickness of 6 millimeters. Such a low thickness can't affect the horizontal components of velocity in a section. That is why the behaviours of U_x and U_y are not changing in both experimental setups. On the other side, the flat plate has a surface area of $975[cm^2]$ and its effect on the flow is absolutely not negligible. First of all the flow is no more axialsymmetric, the symmetry is broken by the presence of the rectangular body on a side of the chamber. In Fig.(5.9) can be seen how the plate as a wall, nullifies the U_z values on the left side of the section. It is interesting to see U_z in Fig.(5.10) on a new plane, normal at y axis. On this section the flow is again symmetric because the plate is centrally disposed. The plate is only a reference in this section because it is not actually intersecting the surface. In general also for the *Plate* setup the *Clockwise* and the *Counterclockwise* cases present similar flows. As happens for the *noPlate* case there is a slight decrease of the values of velocities for the *Counterclockwise* direction of rotation.

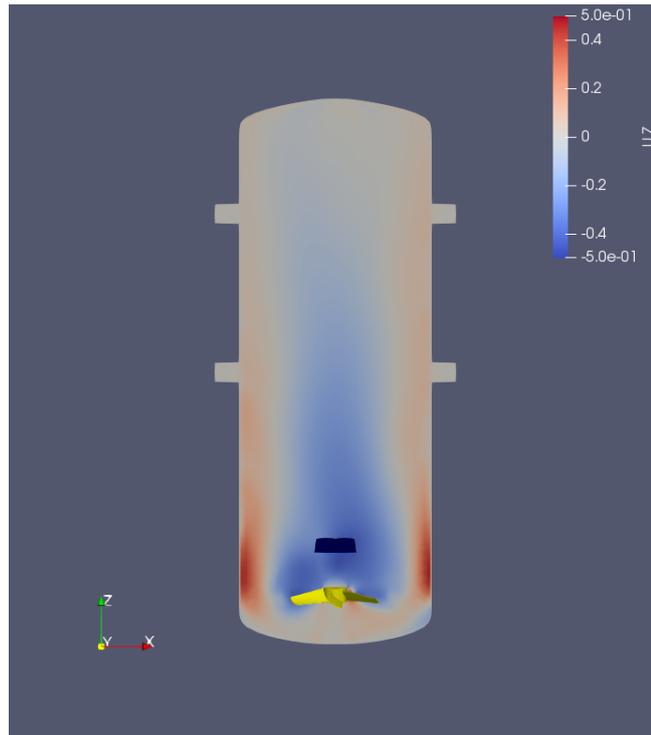


(a) *Clockwise*

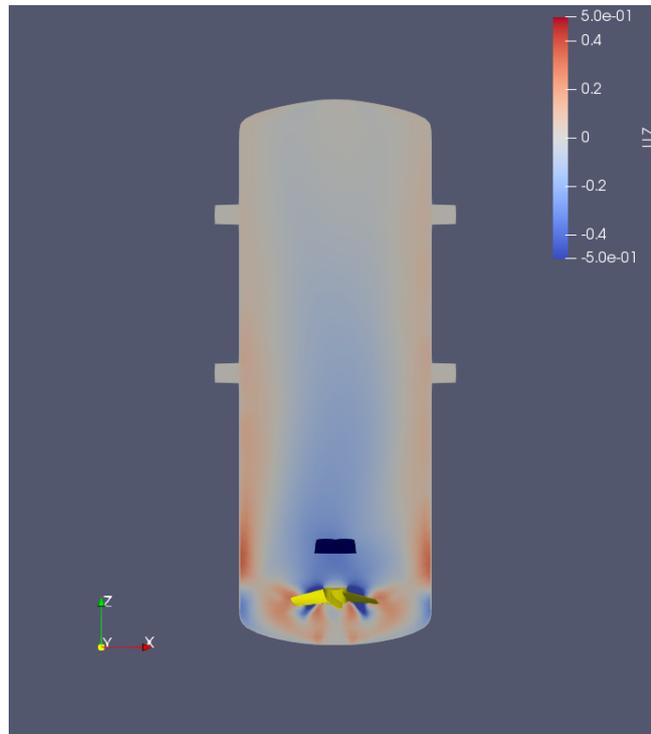


(b) *Counterclockwise*

Figure 5.9: *Contour of U_z on vertical section x – normal*



(a) *Clockwise*



(b) *Counterclockwise*

Figure 5.10: *Contour of U_z on vertical section y – normal*

Results of the Parametric study

For the parametric study of the *Plate* case the following simulations were done:

- *Clockwise* with $f = [1, 3, 5]Hz$
- *Counterclockwise* with $f = [1, 3, 5]Hz$

The total number of simulations is six, half of the previous study. Reducing the simulations and the computational time was a choice made from the following assumptions. It is known that the plate is not affecting too much the flow so similar behaviour of the previous case are expected. After the analysis of the first parametric study we saw few differences in the fields so it is clear that the intermediate cases at $f = [2, 4]Hz$ will not be really relevant. The reason for excluding $f = 7Hz$ is simpler: velocities inside the chamber were too high for the perspective of a particles injection. The results of the simulations have some common points with the *noPlate* case. In order to avoid a redundant section they will be shortly resumed. The cases are relatively insensitive to the direction of rotation. The growing frequency enhance the values of the single components but doesn't change the general tendency of the flow. Profiles of velocity are taken at three different heights in the chamber and in this case at $Height = 0.4[m]$ the profile is between the fan and the plate.

For the *Clockwise* case:

- In Fig.(5.11), profiles of U_x show that the rotational motion of the fluid is conserved at different heights and the values of the component grow with the frequency.
- In Fig.(5.12), profile of U_z show a strong decay with the increasing of height. At $Height = 0.4[m]$ for $-0.4 < y < -0.2$ the presence of the plate can be noticed. The plot is clearly asymmetric.

For the *Counterclockwise* case:

- In Fig.(5.13) U_x is plotted and has the features of the *Clockwise* case with the obvious exception for the direction of rotation.
- In Fig.(5.14) U_z is plotted. When comparing these profiles with *Clockwise* case, the tendency are similar but the values of U_z in this case are lower.

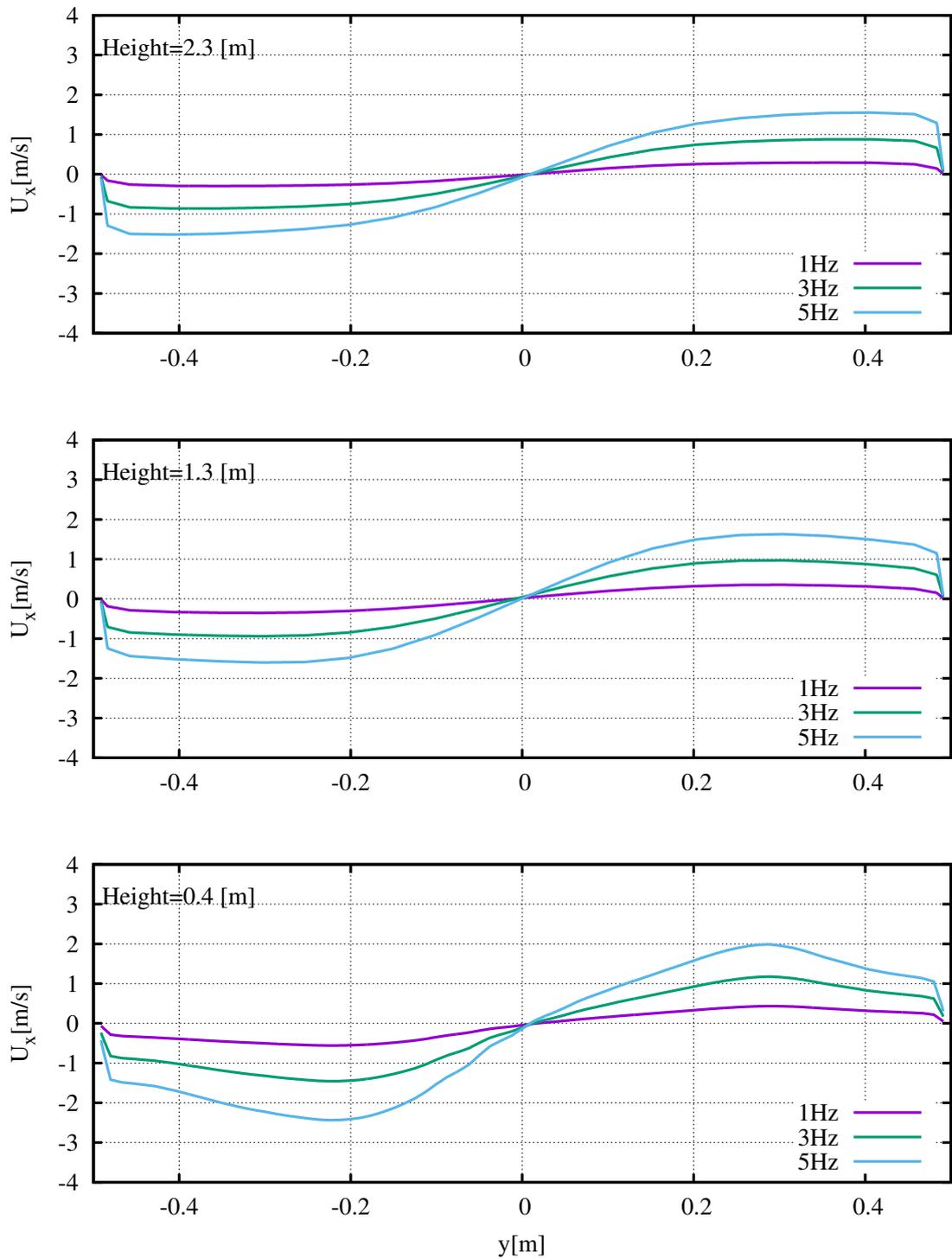


Figure 5.11: Clockwise U_x plots along y at three Heights with f as parameter

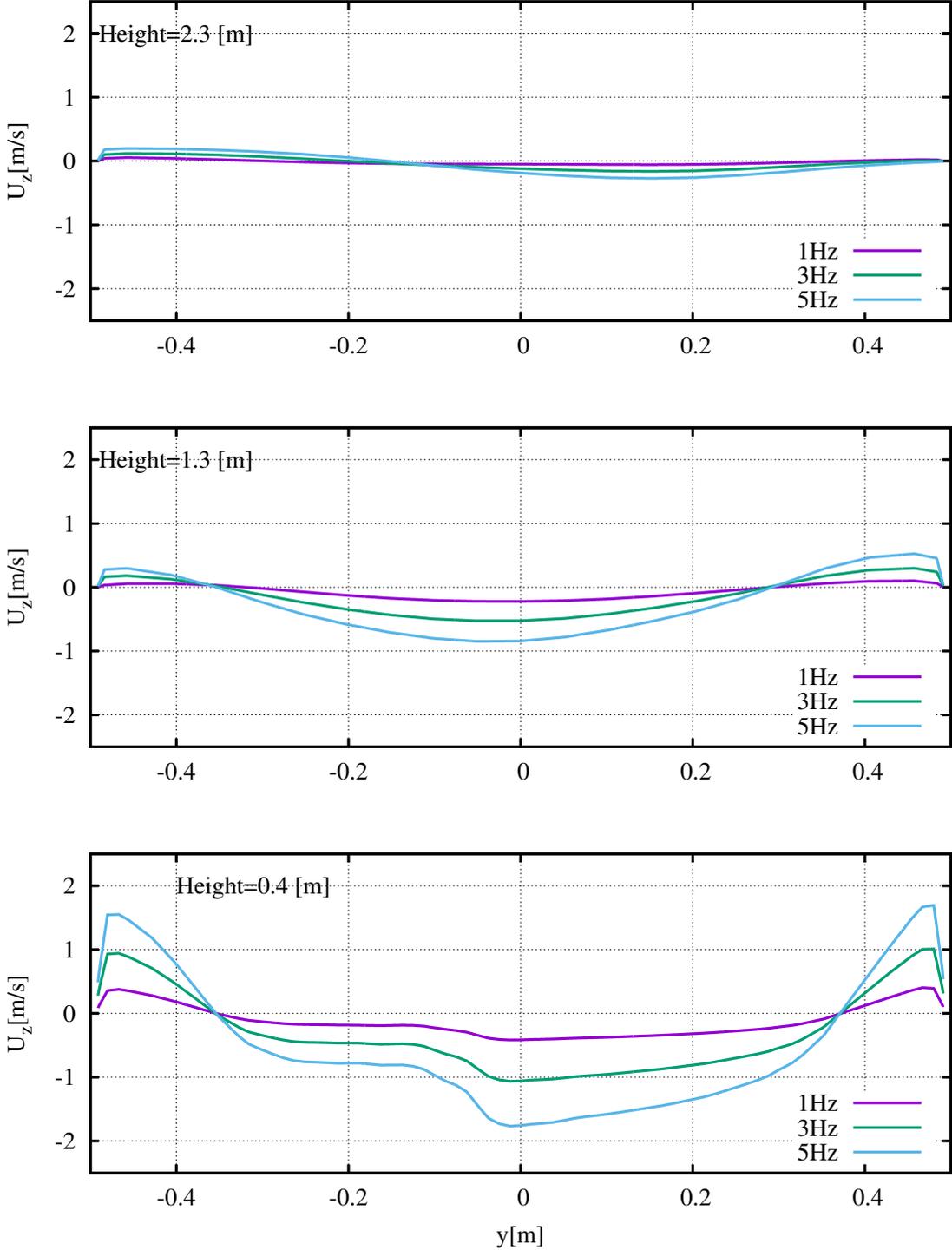


Figure 5.12: Clockwise U_x plots along y at three Heights with f as parameter

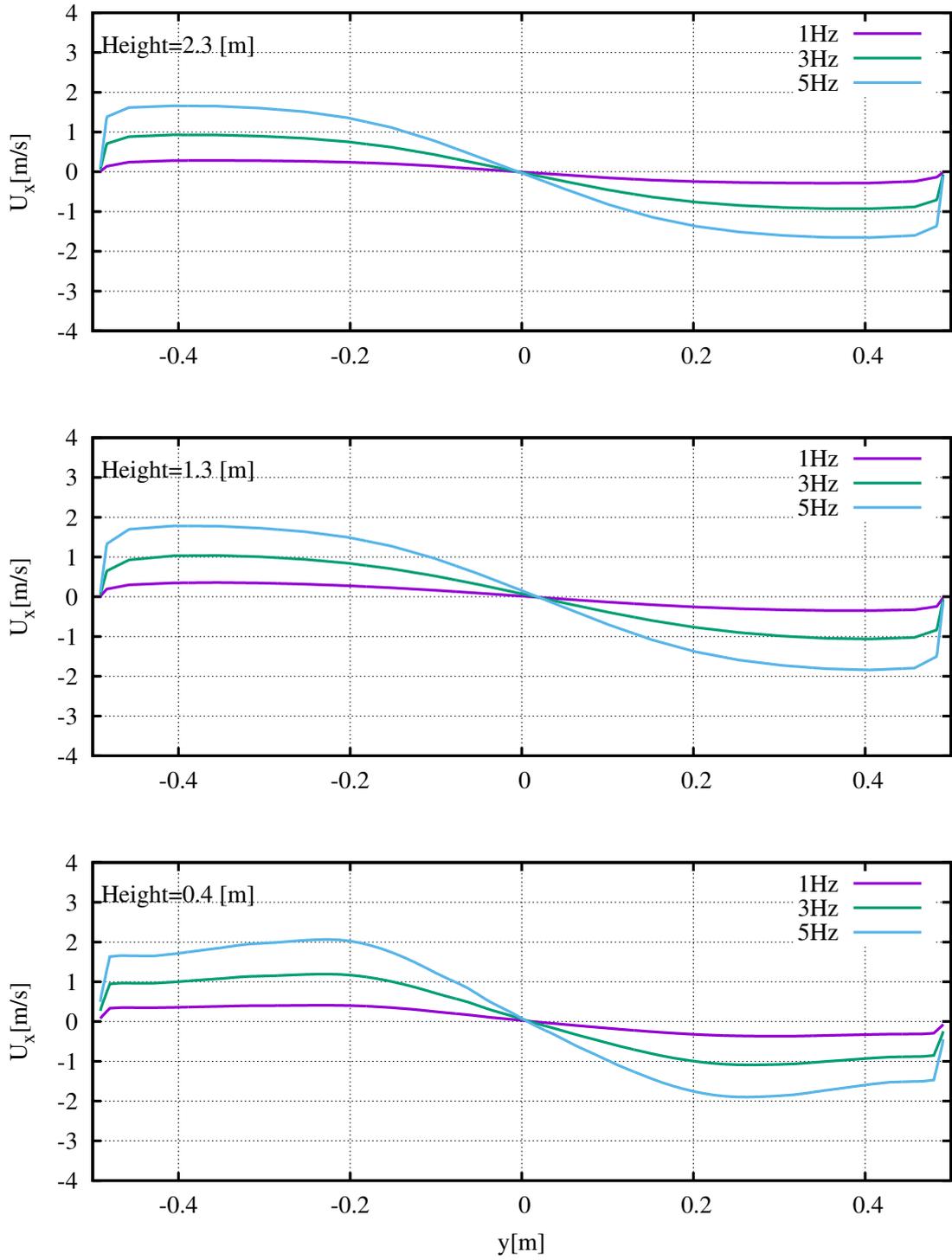


Figure 5.13: Counterclockwise U_x plots along y at three Heights with f as parameter

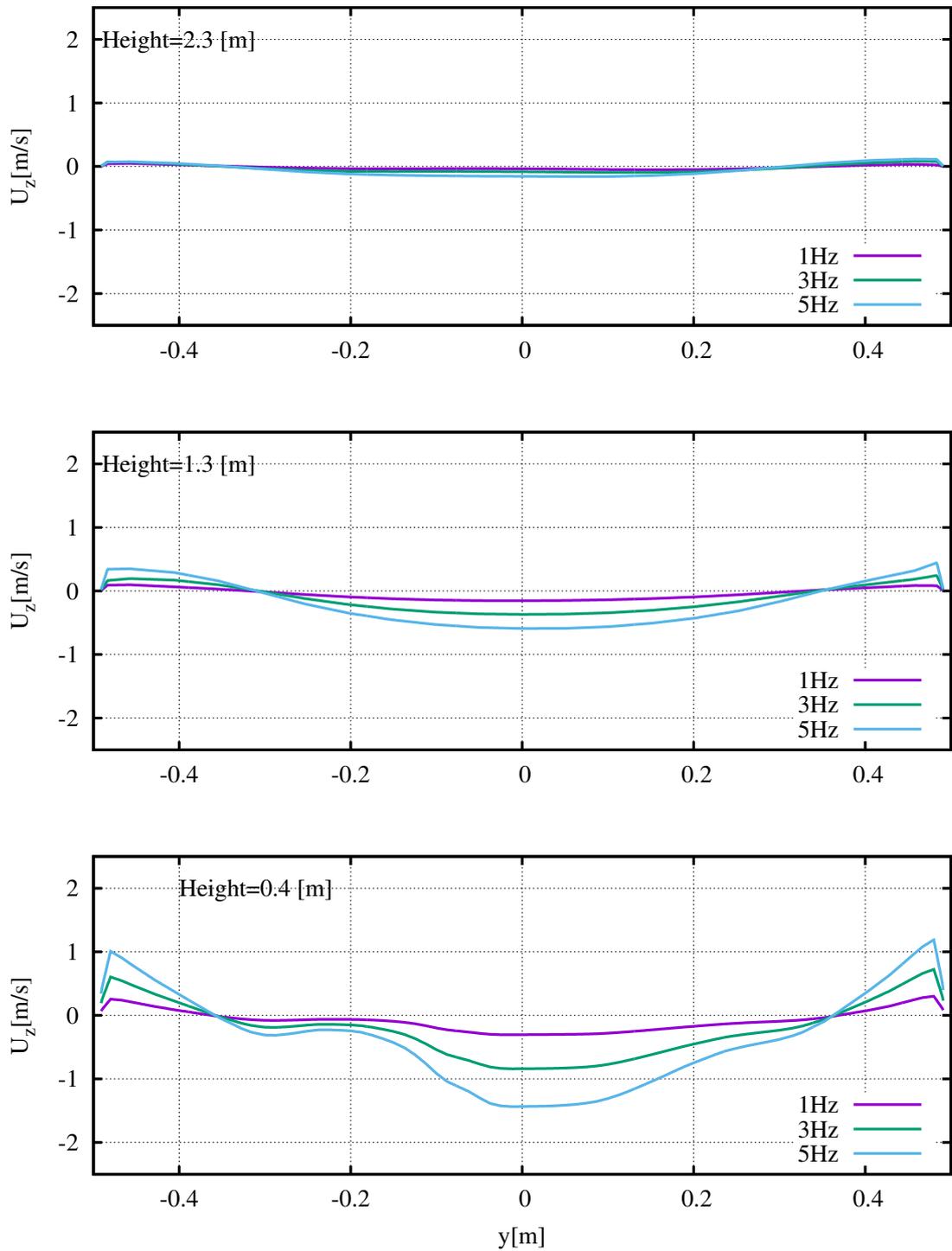


Figure 5.14: Counterclockwise U_z plots along y at three Heights with f as parameter

5.4 *Plate* and *noPlate* cases comparison

In this last section an interesting comparison between the two experimental setups should be done. The terms of the comparison are *noPlate* and *Plate* cases at fan frequency of $f = 3Hz$ and *Clockwise* rotation. With this intent we used a *paraView* function called *streamTracer*. This tool allows the user to visualize the track that a cloud of points will follow in the computational domain. These points are called seeds and they are userdefined. The following images can be misunderstood, it has to be underlined that the tracks represented are not related with the particles cases, these will be seen in the next sections. Both Fig.(5.15) and Fig.(5.16) images are taken by inseminating a thousand of seeds in a central sphere of radius $r = 0.2$ that is surrounding the fan. From this sphere it has to be said that a single snapshot of this view is really limiting if compared with the actual software visualization. What can be seen in Fig.(5.15) is that the tracers for the *noPlate* case are regular, axisymmetric and a huge rotating vortex is descending in the center of ChAMBRé. For the *Plate* case in Fig.(5.16) the flow is asymmetric and more chaotic. The vortex is thinner and disturbed by the presence of the plate.

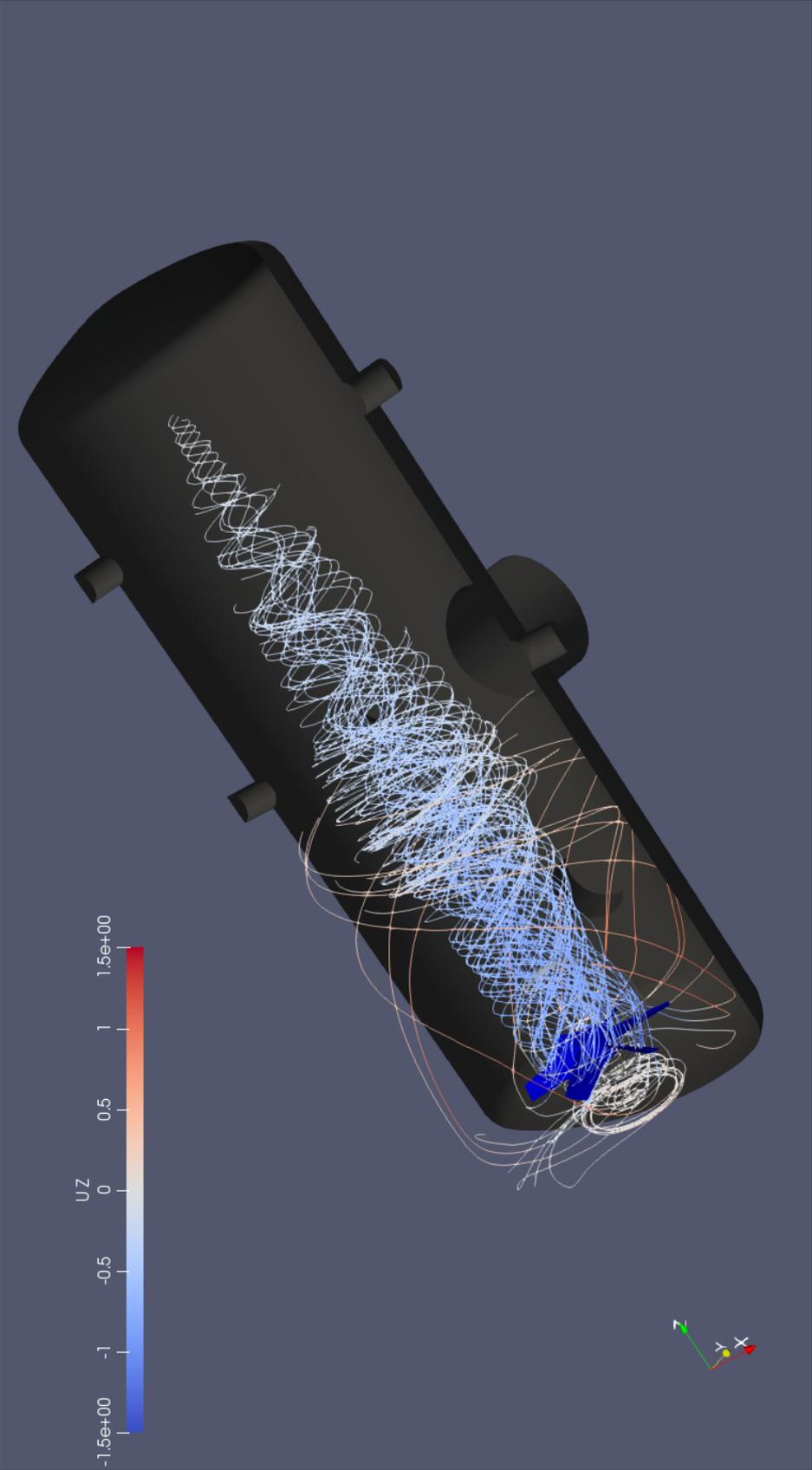


Figure 5.15: *noPlate case Stream tracers at $f = 3Hz$*

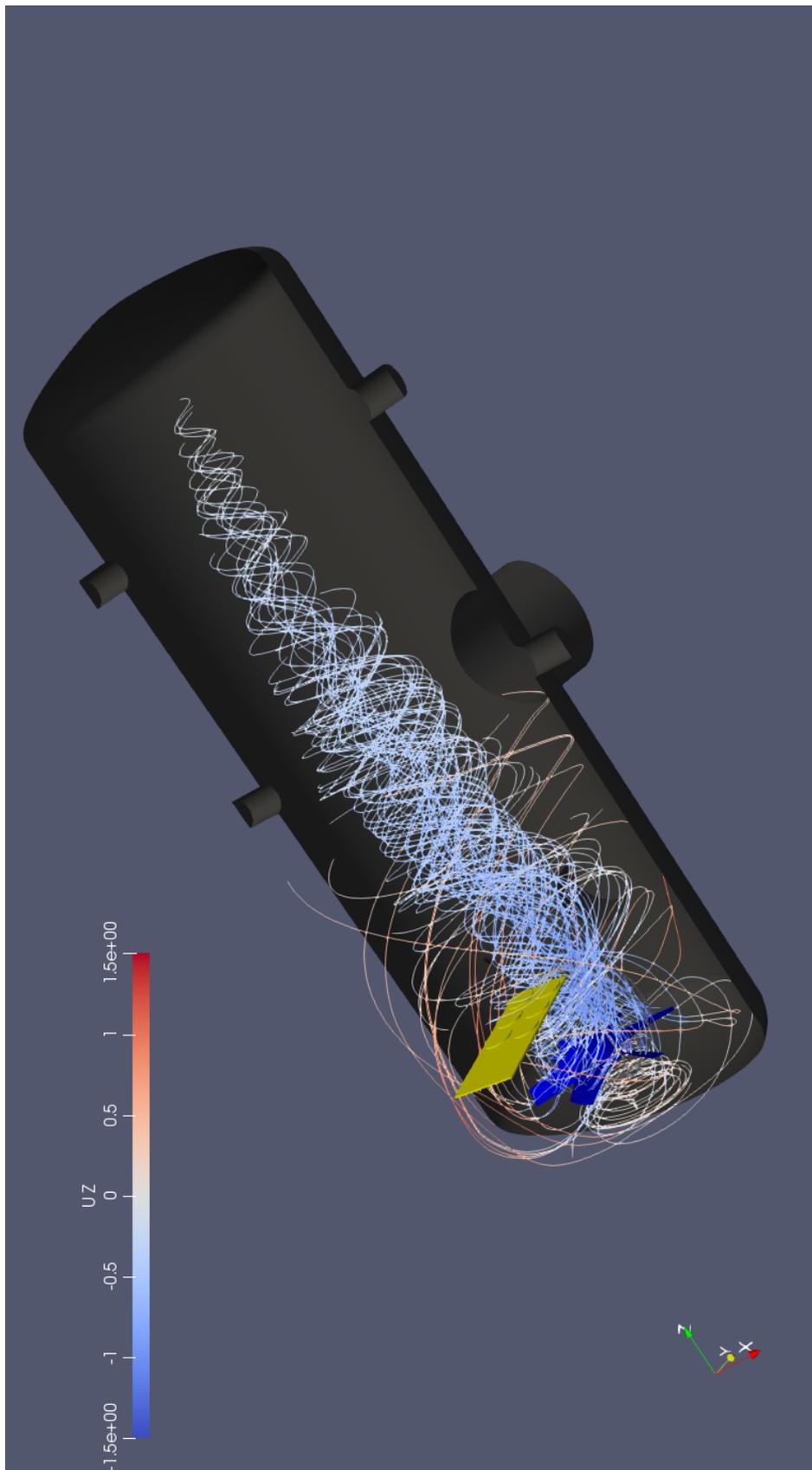


Figure 5.16: *Plate case Stream tracers at $f = 3Hz$*

Chapter 6

URANS Simulation with Sliding Mesh Method

After the first simulations it was clear that a single simulation with a deeper analysis of the flow field could be really important. In the previous RANS cases the solution was time averaged, the following case, URANS, will take into account the transient term of the momentum equation. This approach compared with RANS will be much more time consuming even if URANS is considered the least computationally expensive method to predict the behaviour of unsteady flow. Nevertheless the *Sliding Mesh* method, that actually rotate the mesh, is a way more complex than MRF method. With such an high computational cost, it was clear that a single case should be simulated. Aside from the time factor, running multiple cases will not be a clever choice. It is known from the previous section that, for the several cases, the basic features of the flow will not change much. In the case chosen for a more accurate simulation the ChAMBRé is without plate with the fan rotating clockwise at the frequency $f = 1Hz$. The choice of the parameters was reasoned with the INFN. We decided to simulate this experimental setup because it is one of the most used during the research activities in the facility. Concluding, in the computational enviroment, the configuration for the case is here shortly listed:

- *Fine Mesh*
- URANS Unsteady Reynolds Averaged Navier Stokes equations;
- *SST* $k - \omega$ turbulence model;
- *Sliding Mesh* method;

The followings sections briefly describe the mesh generation process, the *Sliding Mesh* method and the results obtained.

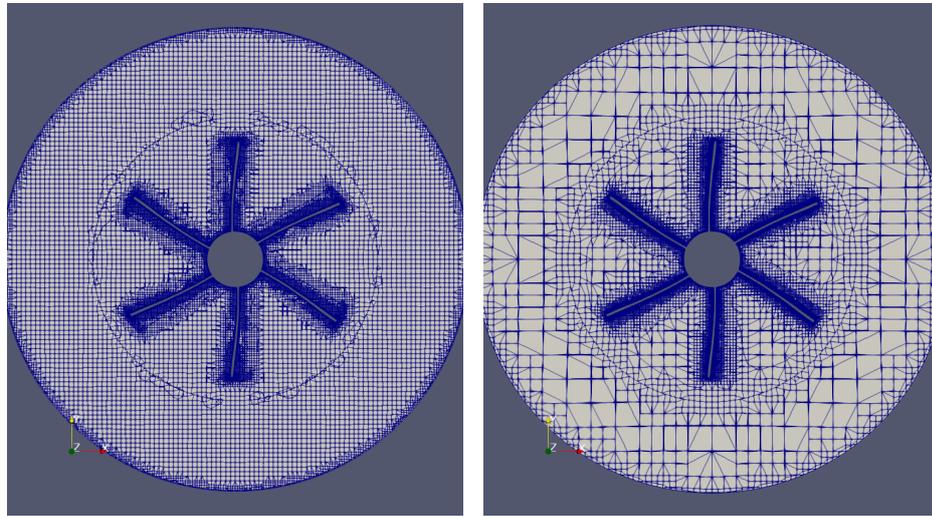
6.1 *Fine Mesh* generation

The *Fine mesh* generation started from the previous *Coarse Mesh* with the intent of obtaining a finer and better mesh. The process has been done with the same OPENFOAM[®] utilities: *blockMesh* and *snappyHexMesh*. For a better understanding we are not going to recall every single step but the section will explain the process through a comparison with the *Coarse Mesh*. First, in the *blockMesh* utility, the spatial discretization has been reduced by half. Hence the background mesh will results a way more fine. When it comes to *snappyHexMesh* utility, having a finer background mesh will corresponds to an increase in all the refinements. The work in this section is more ruled by experience, trial and error. Basically the aim is to find a compromise between mesh quality and mesh dimensions. In particular with high levels of refinement it is needed to contain the number of cells, because this value can easily reach dozens of millions of cells. The sub-dictionary *meshQualityControls* has been used iteratively imposing several parameters in order to find a mesh that can achieve a better quality. It has to be said that asking for too high quality can make the mesher incapable to do so, then it will output the result of previous meshing step in which quality was satisfied. This partial mesh will indicate at which step the mesher is finding problems, then the user will modify the corresponding part in dictionary. When the meshing process is completed the mesh is evaluated with *checkMesh*. As was done for the *Coarse Mesh*, two *Fine* meshes has been done for the two known setups: *noPlate* and *Plate*. In Tab.6.1 are listed the main features of the two meshes.

Table 6.1: *Mesh features*

	Fine Mesh	Fine Mesh with Plate
Cells	8,755,192	9,507,793
HexaHedra	6,462,633	6,882,570
Prisms	345,038	290,248
Pyramids	2	2
Tetrahedra	176	378
Polyhedra	1,910,189	2,293,882
Non-orthogonality	Max= 70.01 Average= 13.13	Max= 70.01 Average= 13.49
Max skewness	2.999	2.999
Max <i>AR</i>	27.25	39.74
Number of layers	1 chamber, 3 fan	1 chamber, 3 fan, 3 plate

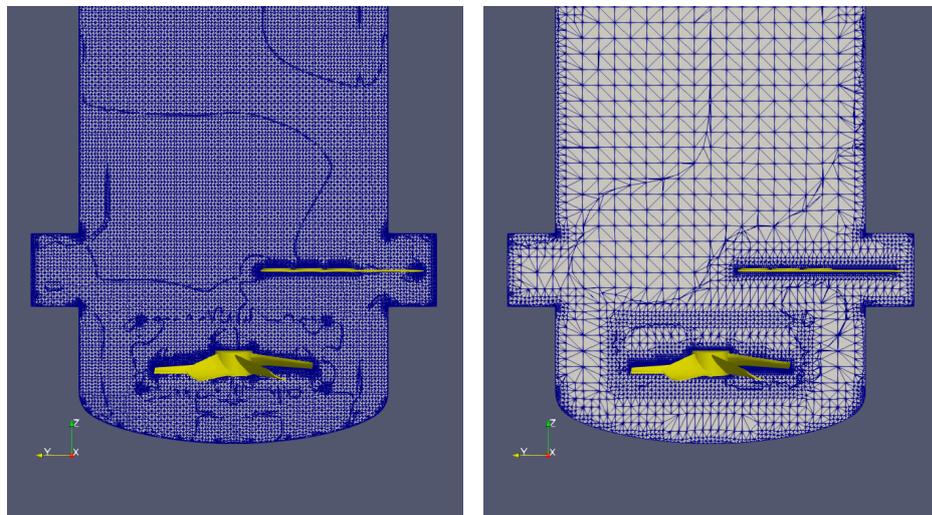
The first parameter that can be noticed is the number of cells that went from about 1.5*million* for the coarse cases to more than 9*millions*. In both *Fine Mesh* we greatly enhanced the refinement on the chamber walls and on the plate. Meanwhile the fan was already well refined but the edges refinement was increased. Furthermore layers were added on plate and on the chamber. In Fig.6.1 the two meshes are visualized on an horizontal plane that cut the fan on his half, the huge increase of cells can be clearly seen. As can be seen in Fig.6.2 the more refined zone surrounding the fan is absent. All the domain inside the chamber is highly and uniformly refined. This feature will make the mesh more compatible with the *Sliding Mesh* method that is going to be shortly described in the next section.



(a) Fine Mesh

(b) Coarse Mesh

Figure 6.1: *Meshes comparison at z normal sections of the fan*



(a) Fine Mesh

(b) Coarse Mesh

Figure 6.2: *Meshes comparison at x normal sections*

6.2 Sliding Mesh Method

The *Sliding Mesh* is a computational unsteady technique to model the CFD problems with rotating parts. The method is completely different from the MRF method and it's capable of dealing with cases with strong rotor stator interactions. The unsteadiness of the model will allow to observe the development of a transient flow field during the time. The idea behind the sliding mesh is too complex and is not covered in this thesis. The OPENFOAM[®] implementation used in this thesis is called Arbitrary Mesh Interface, AMI. It consists in a sliding mesh technique where the rotational zone around the object moves in every timestep and the values lying on the interface are interpolated to update the mesh in every time step. It enables to simulate across disconnected, non-conformal patches but adjacent mesh domains that have been developed based on the algorithm described in [9]. In general, AMI operates by mapping the geometry of one patch on the adjacent one. However, it also projects both patches to an intermediate surface.

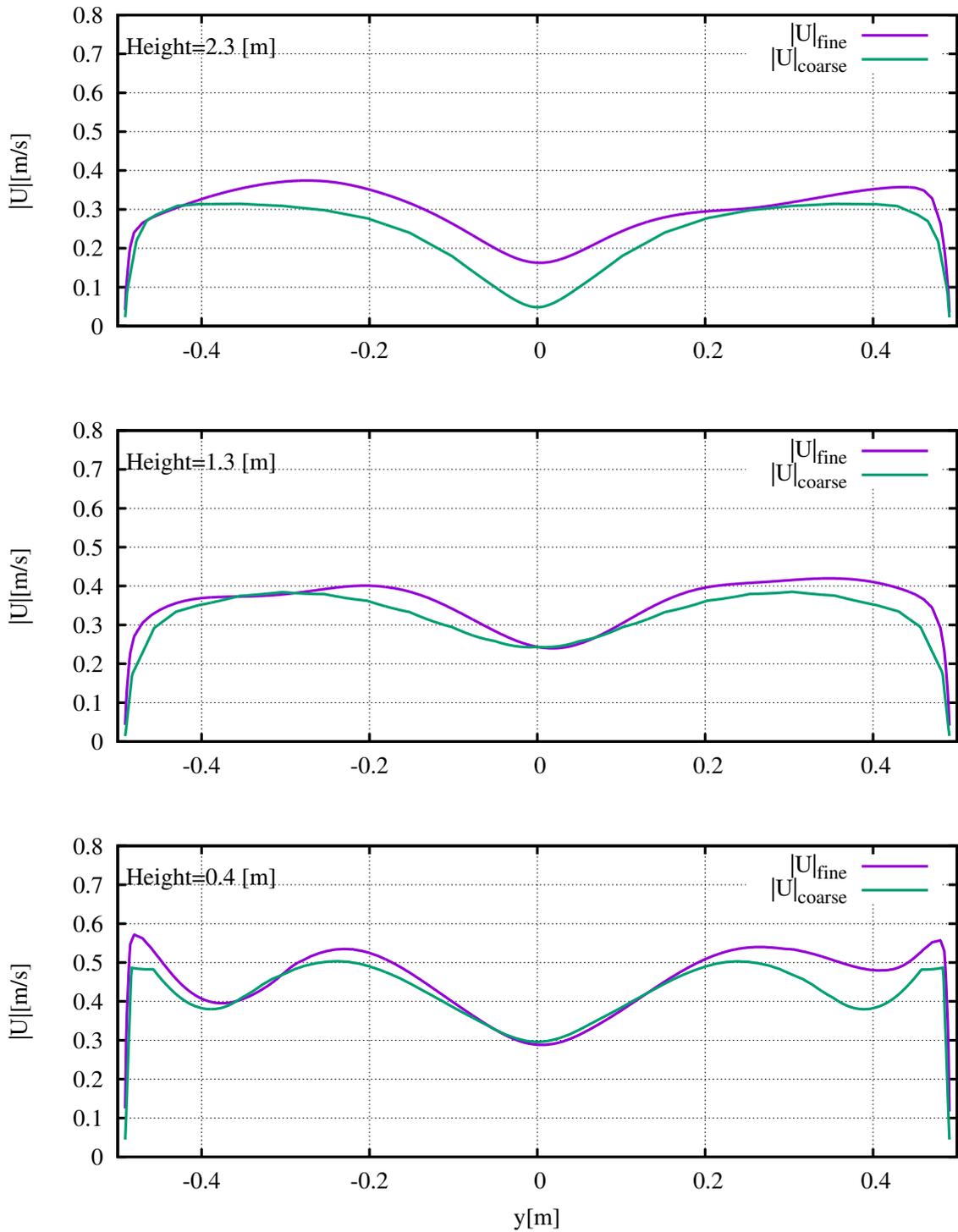
6.3 RANS with *Fine Mesh*

Before moving on the URANS simulation, the obtained meshes were used to run two numerical simulations for *noPlate* and *Plate* cases at $f = 1Hz$. The models and schemes used are the same described in Chapter 5: RANS equations, MRF method and *SST* $k - \omega$ as turbulence model. The basic reason for these computations is that the URANS simulation can be started from zero or from a RANS solved field as initial condition. Initializing an URANS with a computed field, usually, can fasten the convergence and when the time of computation is extremely high, this is really convenient.

The role of these simulations is not only related with this part of the work. In the next part the obtained solutions are going to be the fields in which particles are injected.

Nevertheless it was interesting to analyze differences between the simulations with *Fine Mesh* and *Coarse Mesh* with *noPlate*. It is known that turbulence models are not dependent on cell length. However the comparison between the two cases will show some differences. Recalling that the number of cells in the *Fine Mesh* is nine times bigger than the *Coarse*, it is clear that this new solution will have more spatial resolution. In particular near the chamber walls and inside the flanges the flow is resolved with more detail. In order to have a quantitative view the components

of velocity were plotted at various heights as was done in Chap5. For a single components variation between cases were really small, computing the magnitudes will sum the differeces. For this reason it was chosen to plot the magnitude of U in Fig.(6.3).

Figure 6.3: $|U|$ plots along y for Coarse and Fine mesh Cases

6.4 Results

The URANS simulation with *Fine Mesh* initialized with RANS solution resulted to be extremely time consuming. In order to keep the Courant number constant at $Co = 0.4$ the timestep of the simulation decreased to $\delta t = 7.4e - 06$ so the necessary iterations largely increased. The solution at the moment is obtained for more than one second of real time that corresponds to a complete rotation of the fan. Even if the solution of the flow field is not already converged, few observations on the solution can be done. In Fig.(6.4) the field of velocity U_z is represented for two timesteps. As can be seen the fan is actually moving inside the domain. The solution is currently developing in the higher part of the chamber. Comparing this time dependent simulation with the steady results of Chapter 5 shows that the basic features of the flow are similar. The URANS simulation will give an accurate, unsteady solution that can be the basis for further researches on ChAMBRé.

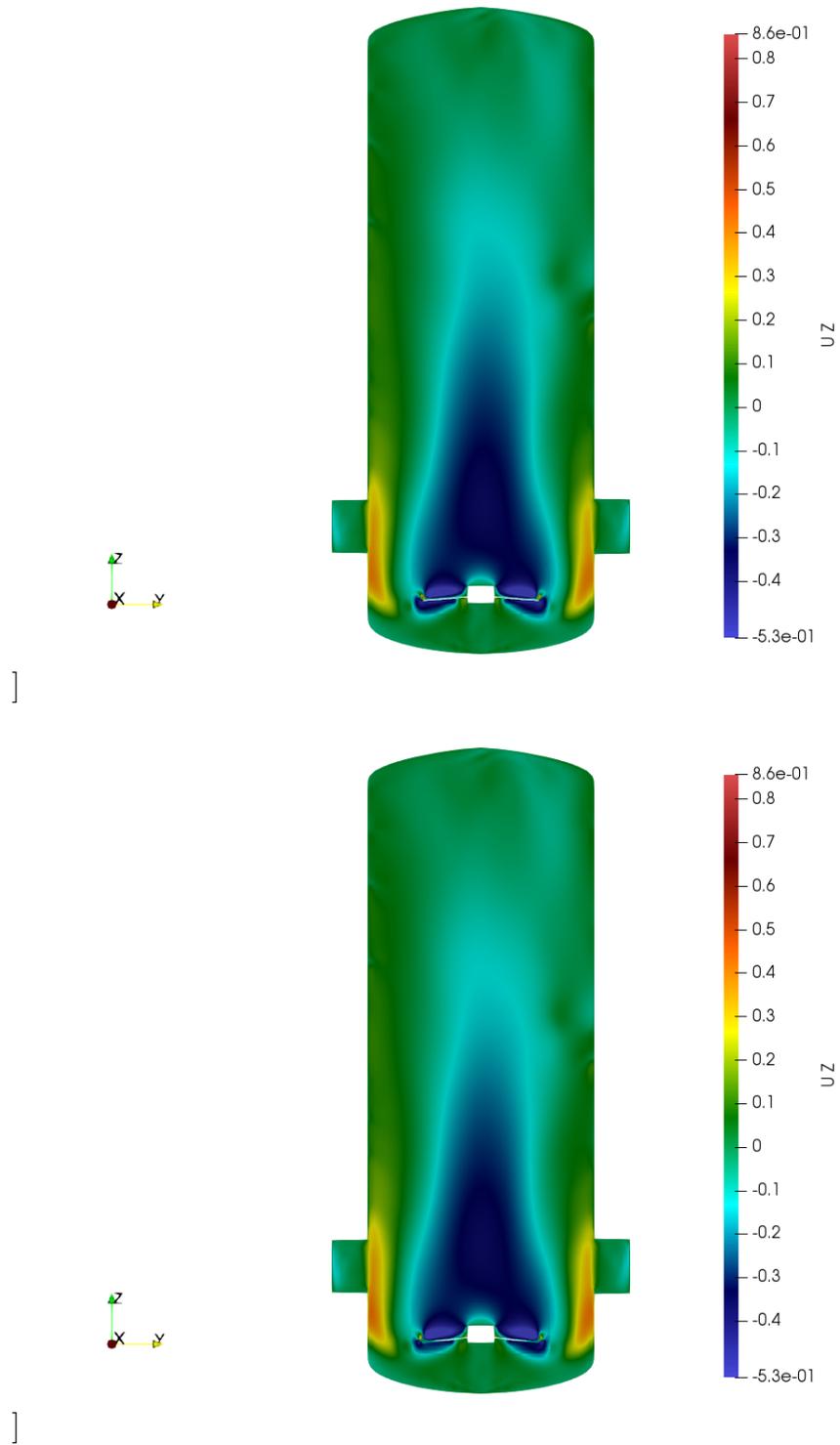


Figure 6.4: Contour of U_z on vertical section at different timesteps

Part III

Simulations with Particles

Chapter 7

Particle Implementation

7.1 Governing Equations

To numerically model a multiphase flow, it is often important to use separate formulations for the different phases. The *particle phase* consists of bubble, particle, or drops and the *continuous phase* is the fluid in which these particles are generally immersed. The particle can be composed of solid, liquid, or gas, and the continuous fluid can be a liquid or a gas. The coupling between the particle motion and its surroundings can be used to classify the character of the multiphase flow, and thus help determine appropriate numerical techniques. The broadest division is between dispersed and dense flows, and refers to which coupling mechanism primarily determines the particle motion. A multiphase flow can be considered dispersed if the effect of particle–fluid interactions dominates the overall transport of the particles, while it is said to be dense if particle-particle interaction dominates particles motion. Dispersed flows includes one-way coupling (where the dispersed-phase motion is affected by the continuous phase, but not vice versa) and two-way coupling (where the dispersed phase also affects the continuous phase through the interphase coupling). Dense flows usually have four-way coupling where mutual interactions between particles become significant and the effects of the particles on the continuous fluid are weak and often neglected. As well described in [12] volume fraction of particles is the main parameter to make the division between one, two or four way coupling and therefore between dispersed and dense. The volume fraction is defined as $\Phi_p = \frac{MV_p}{V}$ where M is the number of particles, V_p is the volume of a single particle and V is the volume occupied by particles and fluid:

- for $\Phi_p < 10^{-6}$ there will be one-way coupling;

- for $10^{-6} < \Phi_p < 10^{-3}$ there will be two-way coupling and the particles can also increase or dump turbulence;
- for $\Phi_p > 10^{-3}$ there will be four-coupling and the flow will be considered as dense.

Another important parameter that may contribute to the selection of the appropriate model is the particle momentum *Stokes number* defined as the ratio between the particle response time τ_p and that of the system τ_s :

$$St = \frac{\tau_p}{\tau_s}, \quad (7.1)$$

$$\tau_p = \frac{\rho_p d^2}{18\rho_f \nu} \quad (\text{only for Stokes flows}), \quad (7.2)$$

$$\tau_s = \frac{L_s}{v_s}. \quad (7.3)$$

If $St \rightarrow 0$, the particle behaves as a fluid tracer (momentum one-way coupling) and if $St \rightarrow \infty$ is unresponsive to the flow variations. One would define a Stokes number not only for momentum but also for mass and temperature in order to evaluate with more precision the mass coupling and energy coupling of particles with the continuous phase. If two-way coupling is considered, it simply involves some source terms in the continuous phase equations (momentum, energy, turbulence models...) that are generally described in an Eulerian reference frame. For the sake of simplicity, we have limited our interest to the one-way coupling.

7.1.1 Particle equations

Various treatments of the particle field can be employed. Particles could be described in an Eulerian or Lagrangian reference frame and, as suggested by E. Loth in his paper [16], distinctions could be done about the treatment of particle surface forces. In this work only the Lagrangian approach will be discussed. With this reference frame the particles are treated as individual and properties are updated along the path of each particle. For the treatment of the surface forces, the point-force treatment represents the flow over the particle with empirical and theoretical treatments (specifying a drag or lift coefficient) to obtain the force on the particle. For the resolved surface treatment, the fluid dynamics (e.g., pressure and shear stress distributions) are fully resolved over the entire particle surface and then integrated to obtain the overall hydrodynamic forces. Following the point-surface approach and defining \mathbf{x}_p as the particle centroid and m_p the particle mass, the Lagrangian

particle equation of motion is:

$$m_p \frac{d\mathbf{v}}{dt} = \mathbf{F}_{body} + \mathbf{F}_{surf}. \quad (7.4)$$

The left hand side represents the particle mass inertia and the right hand side represents the sum of body forces and surface forces on the particle. Body forces are those related to gravitational effects:

$$\mathbf{F}_{body} = \frac{(\rho_p - \rho)\pi d_p^3}{6} \mathbf{g}. \quad (7.5)$$

Where d_p are ρ_p are respectively the droplet diameter and droplet density. Surface forces can be seen as the sum of different terms: drag, virtual mass, a term related to pressure gradient and one to the "history" of particle (Basset term). The expressions for all these terms are listed below without a rigorous derivation:

- The drag force is:

$$F_{D,i} = \frac{1}{2} \frac{\pi d_p^2}{4} \rho_f C_d |\mathbf{u} - \mathbf{u}_p| (u_i - u_{p,i}), \quad C_d = \frac{24}{Re_p} \left(1 + \frac{3}{16} Re_p\right). \quad (7.6)$$

- The pressure gradient force is:

$$F_{P,i} = \frac{1}{6} \pi d_p^3 \rho_f \frac{Du_i}{Dt}. \quad (7.7)$$

- The added mass force (virtual force) is:

$$F_{A,i} = \frac{\pi d_p^3}{12} \rho_f \left(\frac{Du_i}{Dt} - \frac{du_{p,i}}{dt} \right). \quad (7.8)$$

- The Basset force is:

$$F_{B,i} = \frac{3}{2} d_p^2 \rho_f \sqrt{\pi \nu} \int_{-\infty}^t \frac{d}{d\tau} (u_i - u_{p,i}) \frac{d\tau}{\sqrt{t - \tau}}, \quad (7.9)$$

where Re_p is the particle Reynolds number based on relative velocity:

$$Re_p = \frac{\rho d_p (\mathbf{u} - \mathbf{u}_p)}{\mu}, \quad (7.10)$$

where ρ is the density of the continuous phase. A better description of all these terms can be found in [16] and in [11].

7.2 Dispersion Models

Dispersion models are methods that seek to predict how turbulence affects the particles motion. They are employed for RANS simulations with particles in order to account the effect of unresolved turbulent structures. For this reason it is clear that these models will not be used on LES or DNS simulations because the turbulent structures will be filtered and solved or completely solved. The idea behind those methods is to allow particle simulations without the need of a LES or DNS solutions which are extremely time consuming. On the other side these models will have less accuracy and an extremely complex mathematics involving stochastic variables in partial differential equations. In OPENFOAM[®] two dispersion models are implemented: *Stochastic Dispersion Model* and *Gradient Dispersion Model*. The *Stochastic Dispersion Model* adds a stochastic component \mathbf{u}_{turb} at the fluid velocity \mathbf{u} that is seen by each particle. The velocity associated with particles \mathbf{u}_p that determines the forces previously described is computed as it follows:

$$\mathbf{u}_p = \mathbf{u} + \mathbf{u}_{turb} \quad (7.11)$$

The magnitude of \mathbf{u}_{turb} is chosen for each parcel from a Gaussian random number distribution with standard deviation related with the turbulent kinetic energy k . The direction is randomly assigned. A new value of u_{turb} is chosen for a particle after a time related with the local turbulent timescale given by k/ϵ [10].

The *Gradient Dispersion Model* follows exactly the same process of the *Stochastic Dispersion Model*. The difference lies in the assignment of \mathbf{U}_{turb} direction that is not randomly done but it is related with the gradient of k . It should be underlined that even if the code is in terms of $k - \epsilon$, the dispersion models are compatible with other turbulence models as *SST k - ω* which is the one used in this thesis because ϵ can be easily derived from ω .

7.3 Case study, Choice and Setup

After the description of the employed models, the dissertation moves on to the study case. In OPENFOAM[®] all the entries defining the particles are contained in the dictionary *kinematicCloudProperties*. The assignment of proper values is the result of a close cooperation with the INFN staff that provided data and measurements. First of all the choice of the experimental setup to reproduce was not trivial because several types of particles are injected for various purposes inside ChAMBRé.

It is known that the computational effort in particles simulations is strickly related with the total number of particles. For this reason the choice moved on the case with the lowest number of particles; here it concerns to and injection of bacteria. Two types of bacteria are widely studied: *Escherichia Coli* and *Bacillus Subtilis*. This second type can't be simulated because in a stress situation the *Bacillus Subtilis* bacteria tend to gather in groups. Concluding, the reference case is an injection of *Escherichia Coli* with a total number of particles $n_{particles} = 220,000$. This value from a computational point of view represents an interesting opportunity. Often it's not possible to simulate the real number of particles beacause this value can easily be equal to milions or bilions. In this case the experiment can be reproduced with its actual number of particles and experimental data can be related with CFD results more directly.

On the other side such a low number of particles can be confusing and should be claryfied. The complexity of the injection process will induce huge particles loss. As previously described in Chapter 2 the bacteria are mixed in a solution, pumped and aeerosolized. The injection efficiency that is the rate between bacteria injected and bacteria in the initial solution is equal to 1%.

In the simulation the particles will be injected in a flange of the central ring through an orifice with diameter $d = 25[mm]$. The velocity of the entering flow was measured with a hot wire anemometer and resulted to be $U_{in} = 0.05[m/s]$ and reasonably the same velocity was assumed for the particles. During experiments in ChAMBRe the injection lasts for a time $t_{in} = 300s$.

The *injectionModel* of *kinematicCloudProperties* that simulates this condition is called *patchInjection*. The user can define a patch of the mesh and particles will be generated on this patch with the assigned initial velocity.

On the last section of the dictionary *kinematicCloudProperties* the user can define several interactions between particles and patches. Knowing that a bacteria die when they touch a wall, the imposed condtion at walls is called *stick*. As the name suggest, if a particle toches a wall it will remain attached to the wall, the particle will not move anymore ceasing to be *active*. In order to detect and count where the particles will be sticked, every component inside ChAMBRe correspond to a single patch. Another interaction type, *escape*, is imposed at the recirculation patches related with the measure systems. In this case the particle will vanish from the domain. The volumetric flow rate through these patches is really low and will not have a strong effect on the results.

Chapter 8

Particle Simulations and Results

As anticipated in Chapter (6) the particles simulations started from a steady solved RANS field with fan rotation frequency $f = 1$ which is the value used for almost every experiment in ChAMBRé. The solver used for all cases is *icoKinematicUncoupledParcelFoam* which is an unsteady solver for lagrangian particle clouds. It should be underlined that the unsteadiness of the solver will output a time dependent solution for the particles even if the airflow is constant and steady. As this suggest, the coupling will be one-way, only the particles are affected by the gaseous phase and not viceversa. The intial step of the work was to find the proper models for our case. First of all both dispersion models were simulated in cases with *Coarse Mesh* and the *Stochastic Dispersion* model was selected. Succesively, the *Fine Mesh* is the chosen grid for the last simulations for two reasons. On one side, reading the implementation in OPENFOAM[®] for particles tracking, it turns out that, as expected, a finer mesh will grants more accuracy. On the other side, with a pratical approach, it was seen that this finer mesh granted more accuracy without strongly slowing down the computation. In shorts, the followings section will describes two simulations corresponding at the *noPlate* and *Plate* setups using:

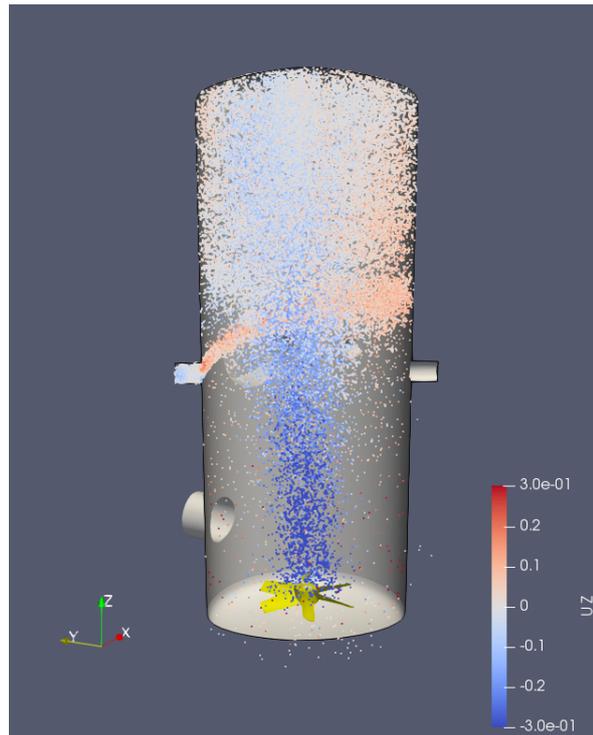
- Number of particles $n_p = 220,000$
- Injection from patch with $U_{in} = 0.05[m/s]$, $t_{in} = 300s$
- For *noPlate* case total simulation time $T_{sim} = 300s$
- For *Plate* case total simulation time $T_{sim} = 400s$
- Volume fraction $\Phi_p = \frac{MV_p}{V} = 5 \times 10^{-14}$
- Stokes number $St = \frac{\tau_p}{\tau_s} = 2.5 \times 10^{-6}$

- Fan *Clockwise* rotation at $f = [1]Hz$
- *Fine Mesh*
- Steady airflow field from RANS
- *Stochastic Dispersion* model

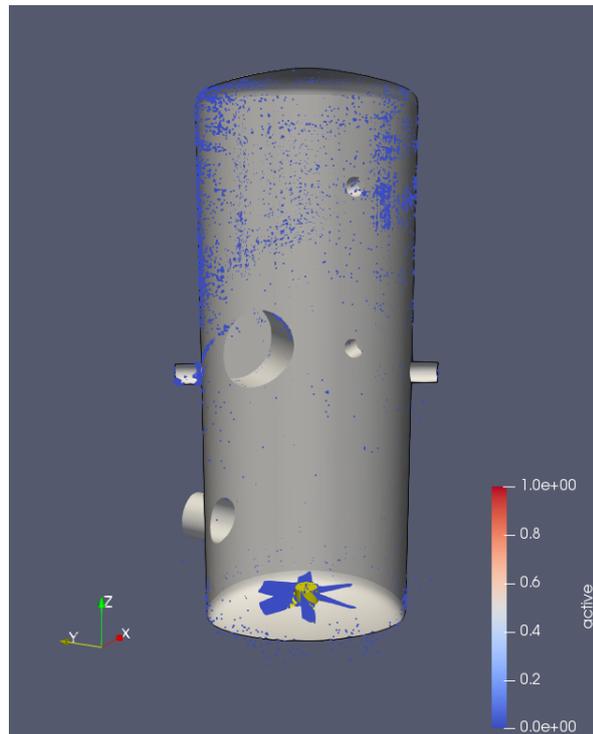
All the cases were visualized using *Paraview*.

8.1 Particles *noPlate* results

The analysis of the results starts from the *noPlate* case. Our interest was focused on the particles fate: where they are going and what is the critical components for the whole experimental structure. As expected the particles follow the airflow and the vertical velocity is determining their fate. After being injected, the cloud passes through the flange and enters inside the chamber. Once entered in the main volume, particles are immediately dragged by the rotational, ascending airflow near the walls. This carries the particles with an initial, spiraliform motion that moves towards the top dome. Near the top of the chamber, vertical velocities are almost zero while the rotation is still present so the cloud is mixed and randomly distributed in the volume. After a few seconds the particles in the central part of the volume start to move downwards. As seen in Chapter(5), the fan generates a central column of descending air. The particles are sucked by this column and pushed into the fan. The simulation has been runned until the end of the injection time $t = 300s$. In Fig.(8.2(a)) at fixed time $t = 300s$ we visualized only *Active* particles. In the cloud a particle is considered active since it doesn't touch a wall. Furthermore each particle is colored with its own vertical velocity U_z , this underlines the general motion: ascending near walls and descending in the center. The qualitative visualization of the *unActive* particles is represented in Fig.(8.2(b)). A particle becomes unactive when it touches a wall. In the experimental reality a unactive particle corresponds to a dead bacterium. From a quantitative analysis it is clear that the fan is the most critical component which is killing more than half of the bacteria. On the chamber wall the stick particles seem to be nicely spread, this means that there are not any critical part or flange that capture a huge number of bacteria.



(a) Active particles vertical velocity U_z

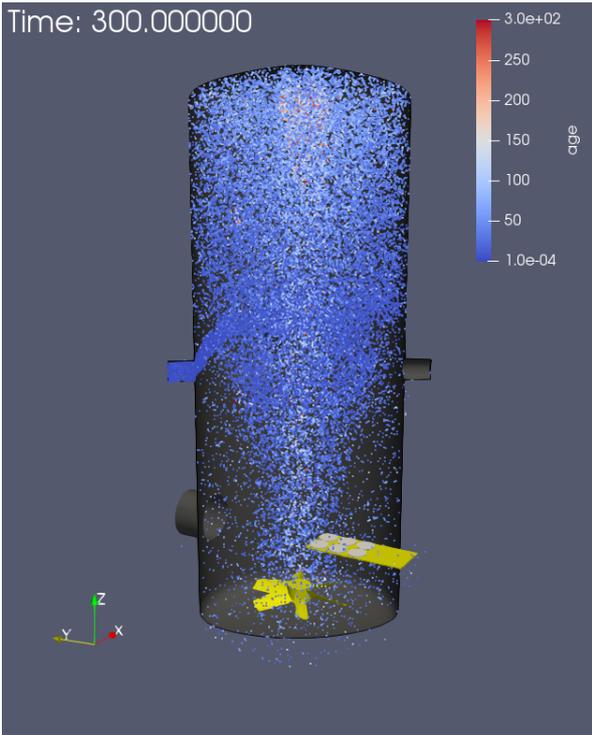


(b) Unactive (stick) particles

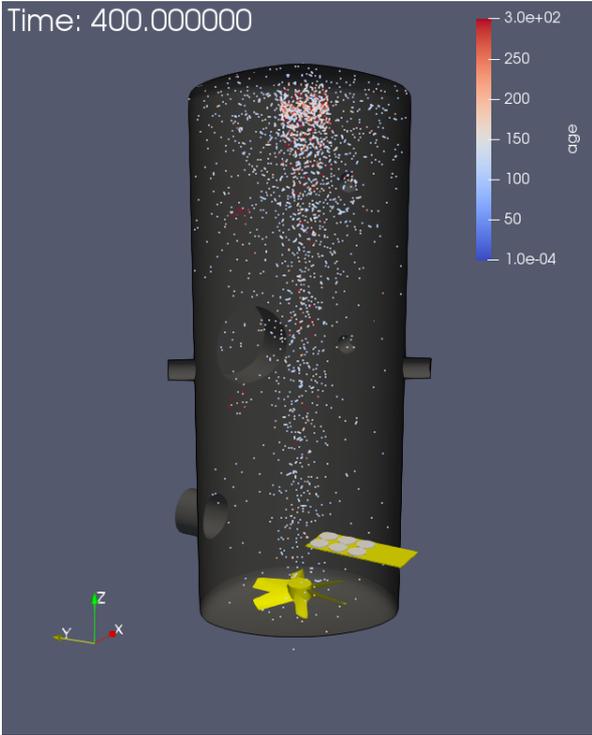
Figure 8.1: *Particles cloud visualized at $t = 300s$*

8.2 Particles *Plate* results

This last case is the most interesting one because the setup with *Plate* corresponds exactly to a real experiment. The same analysis done for the previous case shows that the plate is not changing the basic features of the flow inside the chamber. As it was expected the motion of particles is still governed by the airflow that is similar for both experimental setups. In this case particles are again ascending and rotating immediately after the injection. Successively they are spread in the higher part of the chamber and the column of descending air created by the fan starts to push them downwards. As can be seen in Fig.(8.2) the *Active* particles are almost all vanished from $t = 300s$ to $t = 400s$. During this interval of time the injection has already stopped and the number of *Active* will inevitably decrease. The color of the particles in figure corresponds to their lifetime called *Age*. The *Age* of each particles is the time that passed since the particle was created in the domain. In the higher part, at the center of the main volume particles has the higher lifetime while elsewhere they will barely live more than fifty seconds.



(a) $t = 300s$ Active particles age



(b) $t = 400s$ Active particles age

Figure 8.2

The final part of the analysis has been done using the output data of the solver that counts for each timestep, the number of particles added and stuck to different patches. First of the global number of particles was analyzed. The plot in Fig.(8.3) shows the evolution in time of the number of particles. As can be seen the number of particles in stick condition is plotted for different patches. It is clear that most of particles touches the walls of the fan. The number of particles trapped on chamber walls is relatively low even if the chamber is the patch with the larger surface. While the injection system is working, the number of *Active* particles grows then reaches a stable value. When the injection stops the number of *Active* particles exponentially decreases to zero. There is a huge difference between injected and *Active* particles because a growing number of particles is in *Fan Stick* condition. This confirms what was stated before the fan is consuming a big percentage of the injected cloud. In order to avoid misunderstandings it has to be said that in the key of the plot, the Particles Number means the total number of particles and includes particles that are in stick condition.

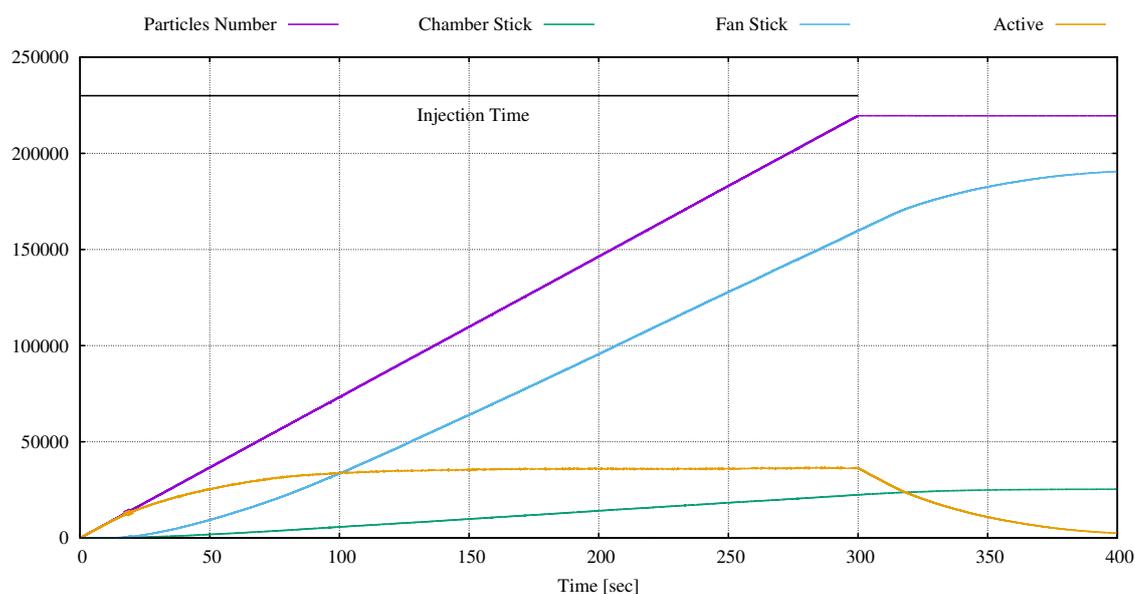


Figure 8.3: *Number of actives and stick particles during time*

The simulation ran for a total time of $T_{sim} = 400s$, this large time interval was chosen because we wanted to compare the results with experimental data. An experiment with this setup usually lasts for five hours but some data were collected also for ten minutes after the starting of the injection. It is clear that five hours of time was far above the computational capabilities so the validation of our models is based on the bacteria number collected on the petri capsules after ten minutes.

On average the INFN staff counted 600 bacterial colonies for each petri after five hour, one colony approximately corresponds to one bacteria for the E.Coli type. In ten minutes they observed that 30% of the colonies are already present. It has to be undelined that this data were collected with four petri capules while in the simulation there are six petri capsules. This was done with the intent of investigate the influence of capsules position on bacterial deposition so two capsules were added on the plate. The number of particles deposited in petri capsules during the simulation time is plotted in Fig.(8.4). The two petri capsules added are *Petri 5* and *Petri 6*. From the plot it is clear that they are almost incapable of collecting bacteria. The position has a fundamental role, infact *Petri 1* and *Petri 2*, the two petri capsules located in the center of the chamber have collected many more particles. After $t = 350s$ the number of collected particles seems to be stable and we don't expect any stornig variation in the values. This tendency was predictable because the *Active* particles are dropping to zero. Since the values are stable it is reasonable to state that the simulation is close to the experimental reality that collected on average 180 bacteria per petri after $T_{exp} = 600s$. Briefly the comparison that gives a partial validation of our numerical simulations is the following:

$$\begin{aligned}\bar{n}_{petri_{sim}}(T = 400s) &= 90 \\ \bar{n}_{petri_{exp}}(T = 600s) &= 180\end{aligned}\tag{8.1}$$

Even if the result of the simulation is the half of the experimental value, it has to be recalled that the total number of particles is 220,000. Looking at the percentages on the total, the validation appears more coherent.

$$\begin{aligned}\% \bar{n}_{petri_{sim}} &= 0.04\% \\ \% \bar{n}_{petri_{exp}} &= 0.08\%\end{aligned}\tag{8.2}$$

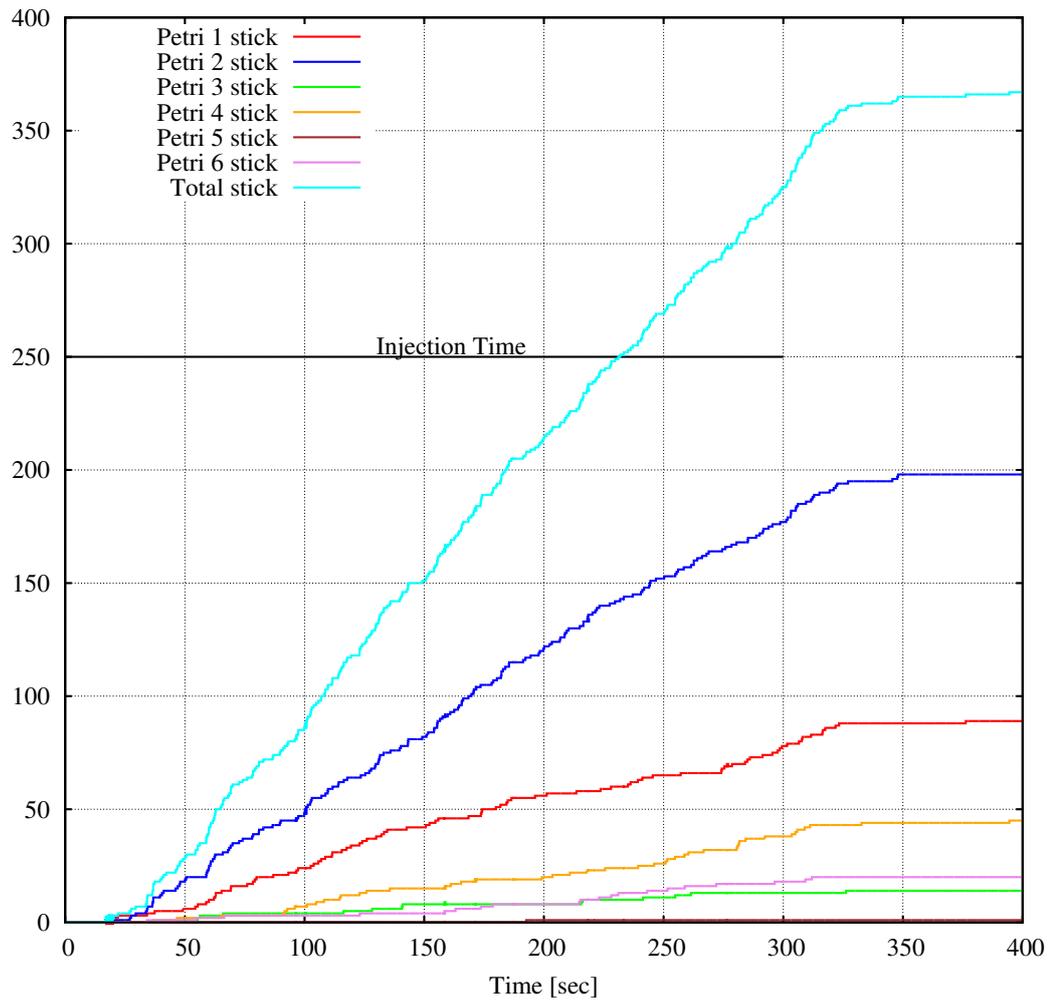


Figure 8.4: *Number of particles stored on petri capsules during time*

Conclusions

The purpose of this master thesis was to study the multiphase fluid dynamics inside ChAMBRé. The results from the first simulations gave a wide overview of the airflow inside the chamber. The parametric study showed that the basic features of the flow field are constant even if the fan rotation is inverted. Nevertheless it was confirmed that the fan always worked in a good operating point. The motion of the air in chamber corresponds to a recirculation that will promote the mixing of aerosols.

The simulations with particles gave interesting results even if several models and approximations have been adopted. Compared to experimental data our simulations are partially validated with a correspondance of orders of magnitude in the number of particles collected in petri capsules. This confirm that the models used are not wrong and the all the assumptions done during the work are adequate. In short it was seen that the fan is killing the most part of injected bacteria. Despite this, the use of the fan is necessary to provide the mixing of particles inside the chamber.

For a computational point of view, future developments are the implementation of unstationary solution with particles in order to model also the transient variations of the multiphase flow. This work is extremely complex as Openfoam doesn't support the parallel computation with particles and rotating elements. Two options are possible: search for other softwares or directly program ad-hoc this capability. Another interesting perspective will be to perform a large eddy simulation on this domain.

With an experimental approach, it could be interesting to run several experiments with the fan that rotates for a limited time. This can enhance the number of collected particles. As it was seen from the simulation the fan is sucking particles that will collide on its blades. For this reason the running time of the fan can be limited to the time necessary to provide a good mixing. Concluding, the presented work marks the beginning of a collaboration with INFN and give a solid base for future studies and developements of ChAMBRé.

Bibliography

- [1] Web page: <https://www.eurochamp.org>.
- [2] Massabò D., Danelli S.G., Brotto P., 2018, ChAMBRé: a new atmospheric simulation Chamber for Aerosol Modelling and Bio-aerosol Research, *Atmospheric Measurement Techniques*, 11, 5885–5900.
- [3] Lai A., Nazaroff W., 2000, Modeling indoor particle deposition from turbulent flow onto smooth surfaces, *Journal of aerosol science*, 31(4):463–476,
- [4] Web page: <https://www.onshape.com/>.
- [5] Web page: <https://cfd.direct/>.
- [6] Web page: <https://www.wolfdynamics.com/>.
- [7] Mehdipour R., 2013, Simulating propeller and Propeller-Hull Interaction in OpenFOAM, KTH Royal Institute of Technology, Master Thesis.
- [8] Web page: <https://www.openfoamwiki.net/>.
- [9] Farrell P.E., Maddison J.R., 2011, Conservative interpolation between volume meshes by local Galerkin projection *Computer methods in applied mechanics and engineering*, 200:89–100.
- [10] Kabanovs A., Varney M., Garmory A., Passmore M., Gaylard A., 2016, Experimental and computational study of vehicle surface contamination on a generic bluff body, SAE technical paper, 2016-01-1604
- [11] Crowe C. T., 2006, *Multiphase flow handbook*, CRC Press, Taylor & Francis Group, Florida
- [12] Elghobashi S.E., 1994, On predicting particle-laden turbulent flows, *Applied scientific research*, 52:309–329.

- [13] Darwish M., Moukalled F., Mangani L., 2016, The finite volume method in computational fluid dynamics, an advanced introduction with OPENFOAM and MATLAB. Springer, United States.
- [14] Greenshields. C.J., Openfoam User Guide
- [15] Liberty J., 1999, C++, SAMS, United States.
- [16] Loth E., 2000, Numerical approaches for motion of dispersed particles, droplets and bubbles, Progress in energy and combustion science, 26(3):161–223.
- [17] Menter F.R., 1994, Two-equation eddy-viscosity turbulence models for engineering applications. AIAA Journal, 32(8):1598–1605.
- [18] Tennekes H., Lumley J. L., 1972, A first course in turbulence, MIT Press, United States.
- [19] Wilcox D. C., 2006, Turbulence modelling for CFD, DCW Industries, United States.

Nomenclature

Notations

α	Turbulent thermal diffusivity
ϵ	Dissipation rate
η	Kolmogorov length
ν	Kinematic viscosity
ω	Dissipation of k
Φ	Volume fraction
ρ	Density
σ	Normal stresses
τ	Tangential stresses
ω	Rotational velocity
F	Force
v	Velocity
d	Diameter
f	Frequency
k	Turbulent kinetic energy
n	Number of particles
p	Pressure
Re	Reynolds number
St	Stokes number
T	Temperature
t	Time
U	Velocity in OpenFoam
V	Volume
x	x axis
y	y axis
z	z axis

Subscripts

<i>exp</i>	experimental
<i>in</i>	injection
<i>p</i>	particles
<i>r</i>	reference frame
<i>sim</i>	simulation
<i>turb</i>	turbulent
<i>x</i>	x axis component
<i>y</i>	y axis component
<i>z</i>	z axis component

Definitions and acronyms

3D	Tridimensional
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
ChAMBRé	Chamber for Aerosol Modelling and Bio-aerosol Research
GNU	General Public License
INFN	National Institute Nuclear Physics
LES	Large Eddy Simulations
MRF	Multiple Reference Frame
OPENFOAM	Open Source Field Operation and Manipulation
RANS	Reynolds Averaged Navier Stokes equations
SST	Shear Stress Transport
URANS	Unsteady Reynolds Averaged Navier Stokes equations
UV	Ultra Violet